Escuela Politécnica Superior

Análisis de la evolución morfológica del cuerpo y los datos clínicos de pacientes en tratamiento dietético nutricional

Máster Universitario en Ciencia de Datos

Trabajo Fin de Máster

Autor:
Nahuel Emiliano García D'Urso

Tutor/es:
Andrés Fuster Guillo

Jorge Azorín López

Septiembre 2021

Universitat d'Alacant
Universidad de Alicante

Análisis de la evolución morfológica del cuerpo y los datos clínicos de pacientes en tratamiento dietético nutricional

Autor

Nahuel Emiliano Garía D'Urso

Directores

Andrés Fuster Guilló

Jorge Azorín Lopez

Tecnología Informática y Computación

Grado en Ingeniería Informática

Escuela Politécnica Superior

ALICANTE, 14 de septiembre de 2021

Si supiese qué es lo que estoy haciendo, no le llamaría investigación, ¿verdad?

Albert Einstein.

Agradecimientos

Me gustaría darle las gracias a mis amigos y familia, que han estado siempre a mi lado para apoyarme en este camino. También me gustaría darle la gracias a todas las personas del proyecto Tech4Diet que me han guiado y enseñado a lo largo de este trabajo de fin de máster.

Resumen

La gran popularización de las cámaras RGB-D ha provocado que se investigue su aplicación en diversos campos. Uno de los ámbitos donde cada vez son más utilizadas las tecnologías 3D es en el sistema sanitario. El modelado 3D basado en cámaras RGB-D es una disciplina de intensa actividad investigadora y cuyos resultados empiezan a consolidarse proporcionando un alto potencial desde el punto de vista de la transferencia investigadora

En este trabajo de fin de máster se plantea un análisis de los datos obtenidos de pacientes en tratamiento dietético nutricional, tanto clínicos como de sus modelos geométricos 3D. Se trata de un trabajo de minería sobre los datos clínicos obtenidos de los pacientes y sus modelos geométricos 3D. El objetivo es especificar un modelo que permita realizar predicciones de la evolución de la geometría y la apariencia del cuerpo humano lo más precisas y versátiles posibles. Para ello, también habrá que afrontar los diferentes retos que supone un trabajo de minería de datos como la obtención de los datos o el preprocesado de los datos.

El estudio de carácter aplicado en el campo médico persigue la mejora de la adherencia de los pacientes en tratamiento dietético. El trabajo está enmarcado en el proyecto de investigación financiado por el ministerio denominado *Modelado y visualización 4D del cuerpo humano para la mejora de la adherencia al tratamiento dietético-nutricional de la obesidad.*

Índice general

Índice de figuras

Índice de tablas

1 Introducción

En este capítulo se expondrá el marco en el que se ha desarrollado este trabajo de final de máster, su motivación y contexto. También, se presentará un estado del arte en el que expondrá la situación actual de las diferentes técnicas actuales de minería de datos y los diferentes métodos actuales para la obtención y análisis de datos extraídos de nubes de puntos 3D que representan cuerpos humanos.

1.1. Motivación y contexto

Ya hace más de dos años que empecé a colaborar en el proyecto de investigación Tech4Diet. Fue en el trascurso de estos años cuando se afianzó mi interés sobre el campo de la visión por computador y sobre como se podía llegar a extraer información de las imágenes con las que se trabajaba.

El proyecto de investigación Tech4Diet cuenta con el apoyo de la Agencia Estatal de Investigación (AEI) y del Fondo Europeo de Desarrollo Regional (FEDER) con referencia "TIN2017-89069-R"perteneciente al programa Retos 2017 en el que su investigador jefe es Jorge Azorín. En este proyecto se busca facilitar el estudio de la evolución morfológica ocasionada por tratamientos de obesidad. Hoy en día, estos tratamientos son muy costosos pero a su vez muy necesarios, ya que los problemas de obesidad o sobrepeso pueden ocasionar enfermedades crónicas como la hipertensión, diabetes tipo II,

cáncer. También pueden ocasionar enfermedades patológicas neurodegenerativas como el Alzheimer o demencias [Fuster-Guilló et al., 2020]

Con Jorge Azorín y Andrés Fuster realicé mi trabajo de final de grado donde el principal objetivo era realizar un estudio sobre el calibrado de cámaras RGB-D. Ya en esa época, me di cuenta de que la obtención de datos, la cual es una de las faces de la minería de datos, no es algo trivial, sino que requiere de un estudio exhaustivo de todas las partes que conforman el sistema.

En este caso el sistema diseñado consta de una red de cámaras RGB-D que obtienen un modelo 3D del cuerpo del paciente. Este proceso de obtención del cuerpo del paciente se realiza en varias sesiones, lo que nos permite poder visualizar y monitorizar el avance del paciente durante el transcurso del tratamiento nutricional. Sobre estos modelos 3D se pueden obtener diferentes medidas 1D, 2D y 3D. Además, se pueden visualizar mediante gafas de realidad virtual la variación de la morfología del cuerpo y las diferentes mediciones tomadas.

Junto a estos datos geométricos del paciente, en cada sesión se le toman datos médicos. Dentro de estos datos encontramos los valores de tensión diastólica y sistólica, el nivel en sangre de glucosa, triglicéridos y colesterol, datos asociados al nivel de musculatura y de grasa del paciente, etc. Por otro lado, los diferentes pacientes han realizado un test psicológico al principio y al final del tratamiento. Este test nos permitirá averiguar si el tratamiento nutricional junto a la visualización de su cuerpo mediante gafas de realidad virtual afecta sobre las habilidades cognitivas del paciente.

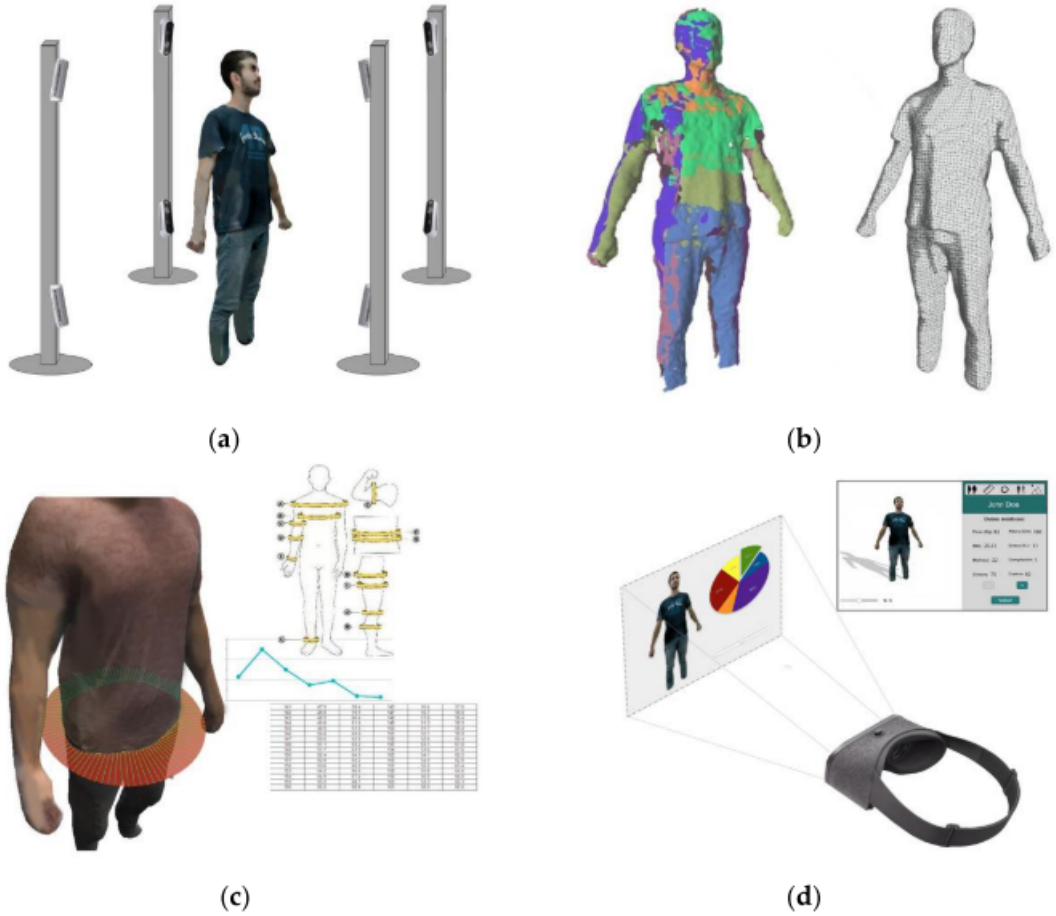


Figura 1.1: Captación del modelo del cuerpo mediante el sistema de cámaras (a). Nubes de puntos sin textura generada (b). Mediciones del cuerpo (c). Visualización de los resultados en la aplicación y con gafas de realidad virtual

Una definición más detallada del sistema de adquisición creado junto a los otros datos recopilados lo podremos encontrar en el capítulo Obtención de datos 2.4. Posteriormente, en el capítulo Tratamiento y preprocesamiento de los datos, se explicarán cuáles han sido todas las tareas aplicadas a los datos para que estos puedan ser utilizados para buscar patrones y crear modelos predictivos. Estas dos últimas tareas se desarrollarán en el capítulo Análisis de los datos. Por último se expondrán las conclusiones obtenidas de todo el trabajo realizado.

1.2. Estado del arte

Como se ha dicho anteriormente, en este apartado se expondrá cuál es la situación actual de las diferentes técnicas en el campo de la minería de datos y cuáles son los diferentes métodos actuales para obtener y analizar datos relacionados con modelos 3D de cuerpos humanos.

1.3. Minería de datos

La minería de datos es una rama de la inteligencia artificial que es aplicada desde los años sesenta [Liao et al., 2012]. En estos años aún no era muy popular el término *Data Mining*, se utilizaban otros como *Data Fishing* o *Data Archaeology*. Durante los años setenta y ochenta la minería de datos se vio beneficiada por el uso de bases de datos relacionales y lenguajes de consulta estructurada (structured query languages, SQL). Fue en esta época cuando el término *Mineria de datos (Data Mining)* se afianzó por la comunidad y se definió como una serie de técnicas y mecanismos, realizados en *software*, para extraer información oculta de los datos. En los años noventa, se reconocía la minería de datos como un subproceso dentro de un proceso denominado Descubrimiento de Conocimientos en Bases de datos (*Knowledge Discovery in Databases, KDD*). La definición más utilizada para KDD es la realizada por Fayyad: "El proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles en los datos" [Fayyad et al., 1996].

Durante los años 2000, fue aumentando el interés por la minería de datos. En el 2003 nos podíamos encontrar con un total de 15 conferencias sobre minería de datos [KDnuggets, 2021]. Durante los siguientes años fue aumentando el número de publicaciones en el campo de la minería de datos como podemos ver en el siguiente gráfico.

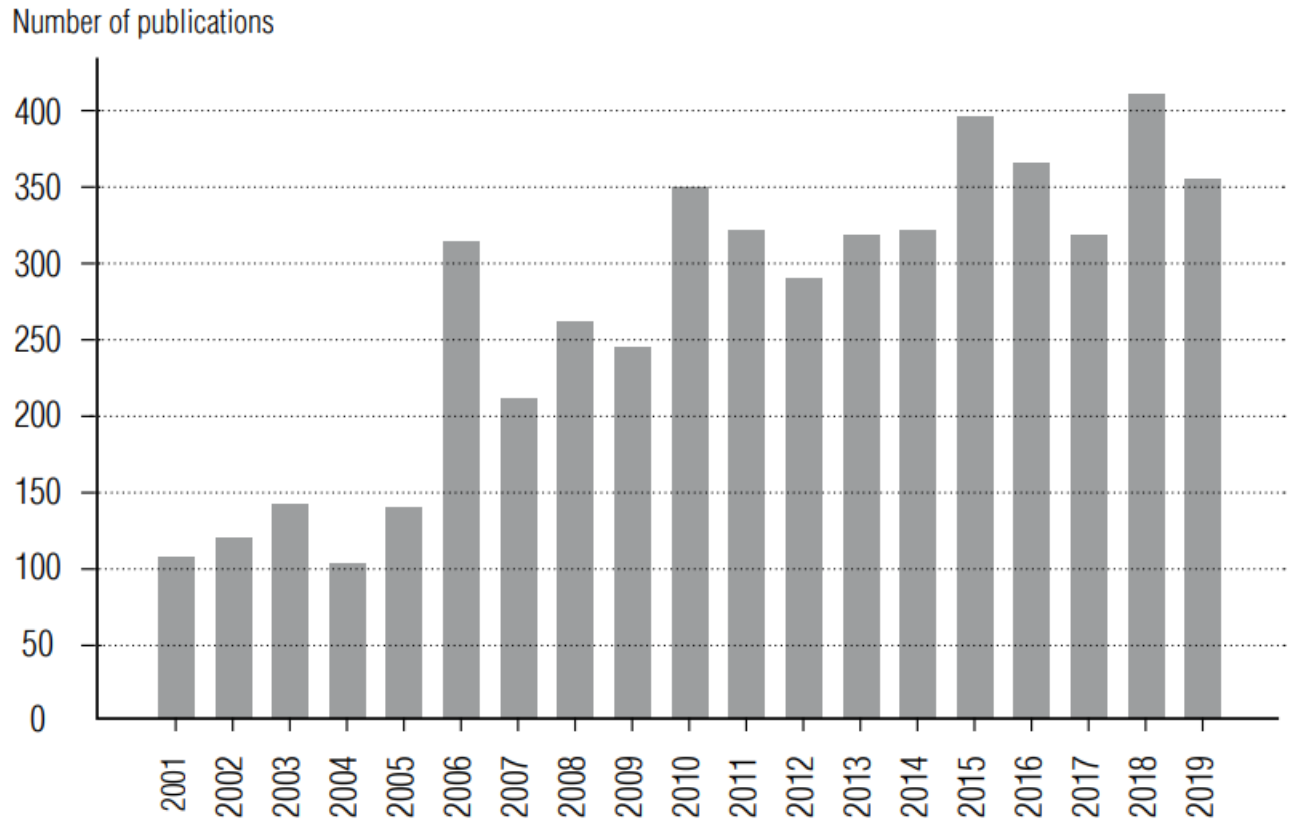


Figura 1.2: Número de publicaciones por año en IEE International Conference on Data Mining (ICDM) [Zelenkov and Anissichkina, 2021]

Durante estos años han sido muchas las herramientas que se han utilizado. Algunas de las más relevantes han sido: [Hosseini, 2021, Jovic et al., 2014]

- **RapidMinier:** Es una herramienta basada en Java que se puede utilizar en Mac OS, Linux y Windows. En su versión 5 o inferior era open source pero desde la versión 6 a pasado a ser una herramienta basada en licencias (Starter, Personal, Professional y Enterprise). La version Starter nos proporciona 1 GB de espacio, pero hay que tener que todo los inputs de datos deben estar en formato CSV o Excel. Permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico [Rap, 2021].

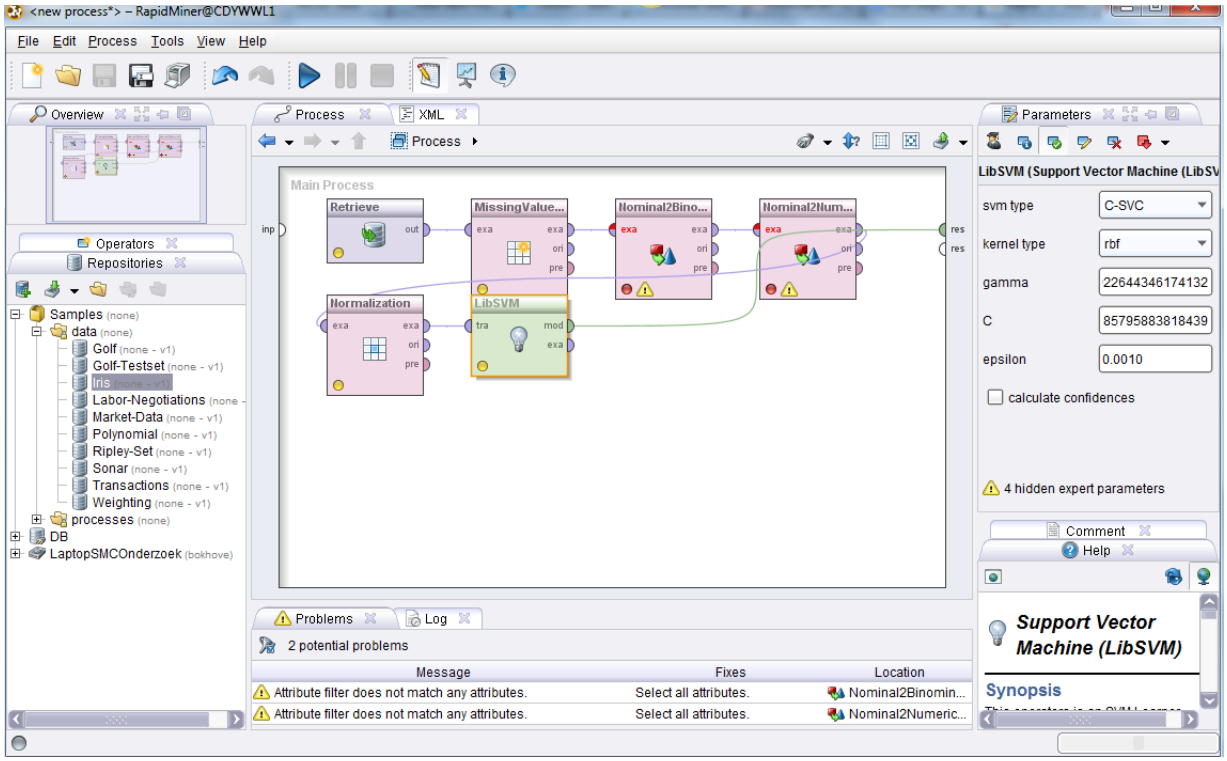


Figura 1.3: Herramienta Rapid Miner

- **Weka:** Al igual que Rapid Miner esta herramienta también está desarrollada en Java. Creada por la Universidad de Waikato y completamente open source. Dispone de una gran variedad de algoritmos de minería de datos. Weka proporciona cuatro opciones para la minería de datos: command-line interface (CLI), Explorer, Experimenter, Knowledge flow y Workbench. Siendo la opción Explorer la más utilizada, ya que nos permite definir cuál va a ser el input, su procesamiento, seleccionar los algoritmos a utilizar y su visualización

- **Orange:** Herramienta basada en Python desarrollada por el Laboratorio de Bioinformática en la Facultad Computación y Ciencias de la Información en la Universidad de Ljubljana. Dispone de una interfaz de selección de tareas aunque también soporta scripts en Python. Algunas de las funcionalidades que podremos encontrar son: operaciones con los datos, visualización, clasificación, regresión, evaluación y

entrenamiento no supervisado [Ora, 2021].

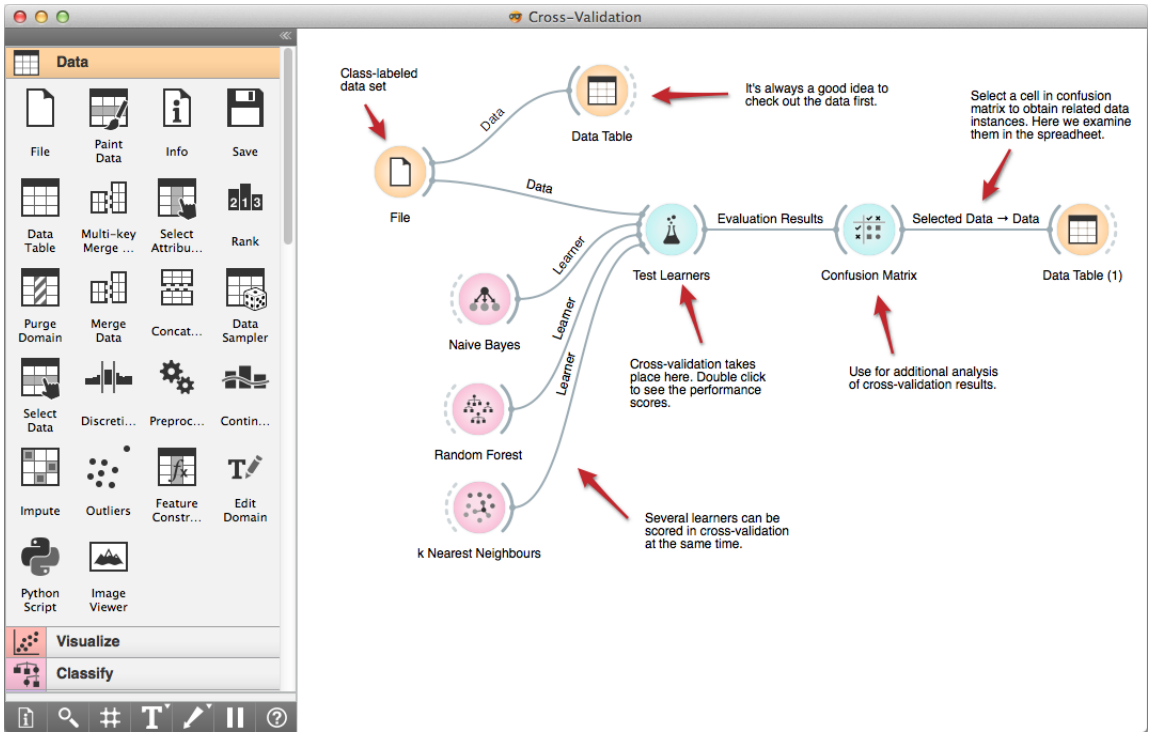


Figura 1.4: Herramienta Orange

- **KNIME:** Construido bajo la plataforma Eclipse en Java esta es una herramienta gráfica que dispone de una serie de nodos que encapsulan diferentes algoritmos. Encontraremos funcionalidades como: manipulación de filas o columnas, visualización, creación de modelos estadísticos y de minería de datos, validación de modelos, scoring y creación de informes [kni, 2021].

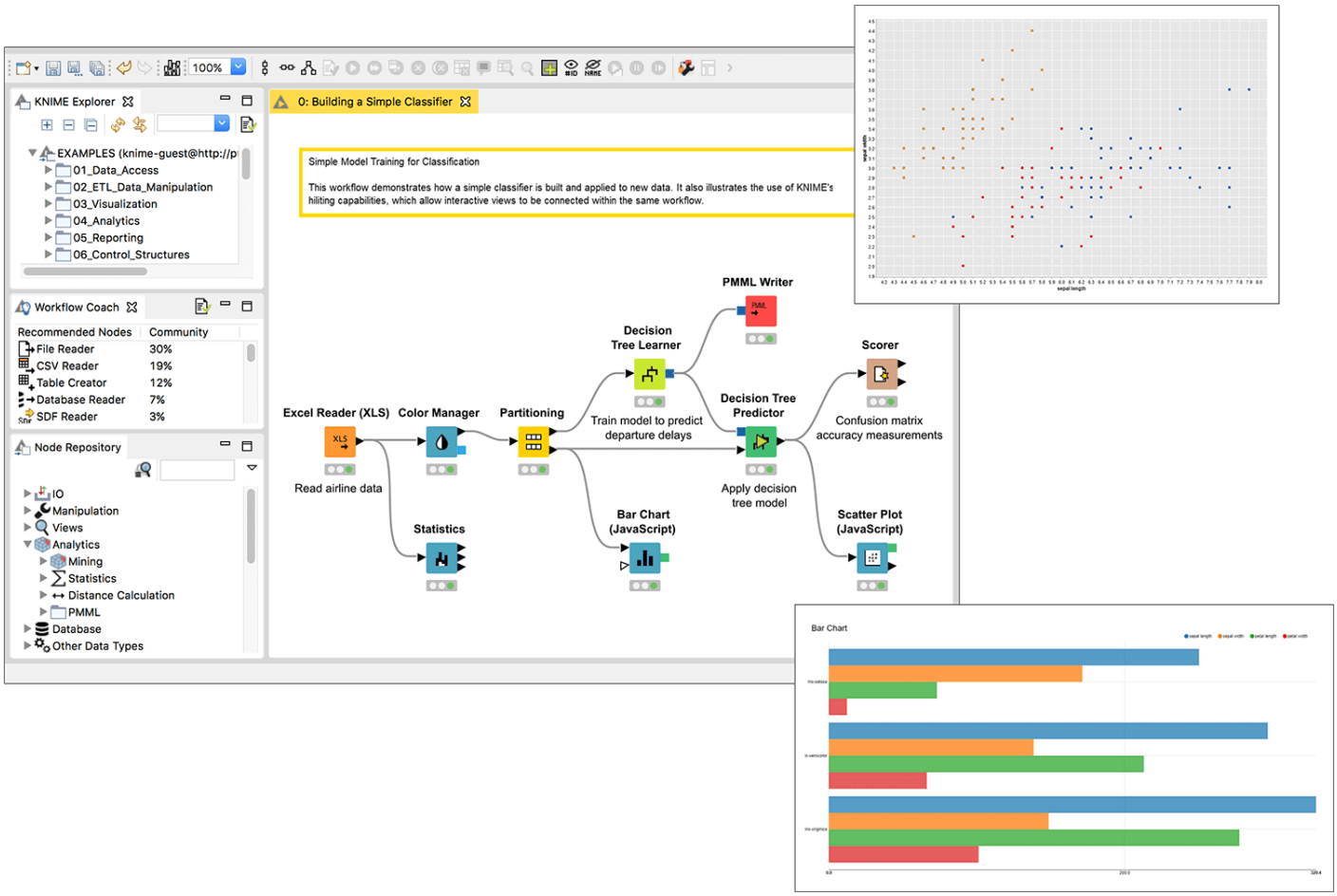


Figura 1.5: Herramienta KNIME

R: El lenguaje de programación R ha sido la principal opción para realizar tareas de minería de datos por estadísticos y matemáticos a lo largo de los últimos 15 años. Creada por Bell Labs en 1970. Es un lenguaje interpretado y mayormente optimizado para realizar cálculos sobre matrices comparables a otras opciones comerciales como MATLAB. R ofrece implementaciones de muchos algoritmos de machine learning. Dispone de tipos de datos específicos para big data, soporta la paralelización, data streams, web mining, graph mining, spatial mining, y otras tareas [r, 2021].

- **IBM SPSS Modeler:** Es unos de los software de IBM para la minería de datos. Permite crear modelos predictivos para la minería de datos y el análisis de textos [ibm, 2021].

- **Azure Machine Learning:** Desarrollado en un entorno basado en la nube y completamente escalable lo que permite a los usuarios crear de una forma sencilla modelos. Es compatible con MLflow, Kubeflow, ONNX, , Python, R, XGBoost, DASK y otros [azu, 2021].

- **Keel:** Herramienta open-source desarrollada en Java. Está centrada principalmente en la implementación de aprendizaje evolutivo y soft computing basado en técnicas de minería de datos como la regresión, clasificación, clustering y pattern mining entre otras [kee, 2021].

Sin embargo, en estas dos últimas décadas todas estas aplicaciones han empezado a utilizarse cada vez menos debido al aumento masivo del uso de Python. Python se ha consolidado como la principal herramienta para todas las tareas de computación científica, incluyendo el análisis y visualización de grandes datasets. Lo que hace a Python una de las mejores herramientas a día de hoy para la minería de datos son todos sus paquetes externos, donde los más populares son:

- **Numpy:** Paquete diseñado para la manipulación de arrays.

- **Pandas:** Librería especializada en el manejo y análisis de estructuras de datos.

- **SciPy:** Librería destinada a tareas de computación científica.

- **Matplotlib:** Diseñada para la visualización de los datos

- **IPython:** Permite una ejecución interactiva del código escrito en Python.

- **Scikit-Learn:** Cuenta con algoritmos de regresión, clasificación, clustering, reducción de la dimensionalidad, etc.

Algunos de los campos más relevantes a día de hoy donde se aplica la minería de datos son:

- Medicina: Un ejemplo del uso de la minería de datos en el campo de la medicina es el que se hace en [Miao et al., 2017], en este trabajo se utilizan técnicas de regresión junto a algoritmos genéticos para estimar la presión en sangre. También se ha utilizado la minería de datos para desarrollar mejores diagnósticos y tratamientos [Koh and Tan, 2005]. En [Mokharrak et al., 2012] se utiliza la minería de datos para detectar el patógeno y examinar el patrón de resistencia a los medicamentos. También, es útil para la clasificación y predicción de múltiples enfermedades [Saeb et al., 2018]

- Finanzas: La minería de datos es usada en este campo para tareas como la previsión del mercado de valores, predecir el ratio de cambio de moneda, quiebras bancarias, trading futures, calificación creditaria, gestión de préstamos, crear perfiles de clientes bancarios o análisis de blanqueo de dinero. [Kovalerchuk and Vityaev, 2005]

- Educación: Educational Data Mining se está convirtiendo en un campo cada vez más importante dentro de la minería de datos. El auge de páginas online para aprender ha hecho que se realicen tareas de minería de datos para averiguar cuáles son los cursos que más le pueden interesar a una persona. Para predecir el resultado de un examen por una persona [Tomasevic and Vraneš, 2019]. Descubrir nuevos patrones de comportamiento de estudiantes [Álvaro Jiménez Galindo, 2010].

- Industria: La minería de datos es usada en varias áreas de la ingeniería de fabricación (manufacturing engineering). Por ejemplo, para monitorizar procesos complejos, Zhang Yingwei y Zhang, Yang utilizan Partial Least Squares con regresión para realizar esta tarea. Por otro lado, podemos encontrar avances donde se utili-

za multivariate linear regression para predecir la calidad de los sensores en vinos [Aleixandre-Tudó et al., 2016].

1.4. Creación de modelos 3D

Con respecto a la creación de modelos 3D de personas humanas y la obtención de datos pertenecientes a los cuerpos nos podemos encontrar varios avances que se han realizado en el Instituto de Sistemas Inteligentes de la Universidad de Max Planck.

En 2015 publicaban el artículo **SMPL: A Skinned Multi-Person Linear Model** [Loper et al., 2015], donde presentaban un modelo capaz de aprender la forma del cuerpo humano y la variación dependiente de la postura. Los resultados mejoraban la precisión con respecto a otros modelos anteriores como BlendSCAPE.

En el 2016, se presentó el artículo **Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image** [Bogo et al., 2016]. En este artículo se partía de una imagen RGB de la persona y con el uso de redes CNN se obtenían los Joints. Son representaciones en el espacio de diferentes puntos del cuerpo. Utilizando un *template*, que no es más que un cuerpo del cual se conocen todos sus parámetros, se ajustan esos puntos 2D hasta obtener la misma posición de la imagen en el *template*. Luego, se ajusta el *template* para estimar la forma (shape) de la persona. Por último, se realiza una última minimización sobre el shape y la pose de la persona. En la imagen 1.7 se puede observar como a partir de una imagen se obtiene el cuerpo de la persona.

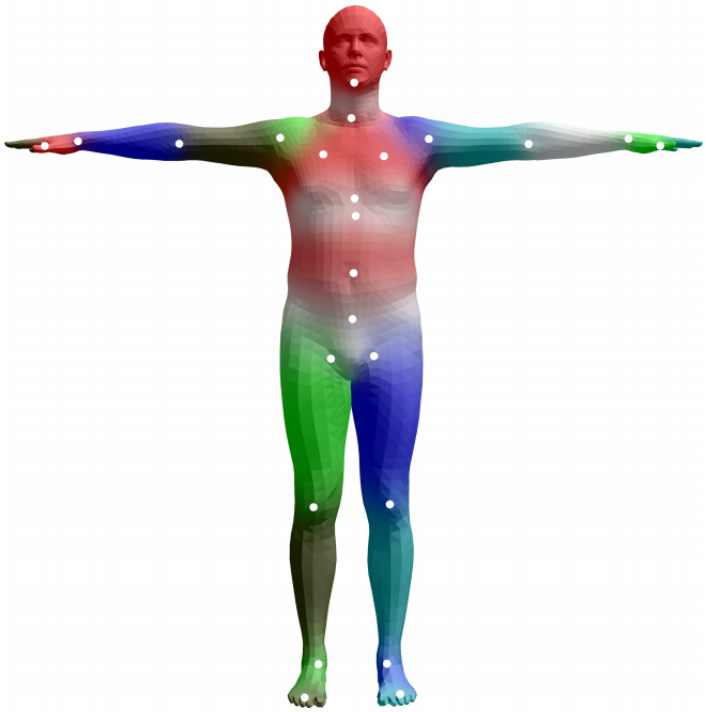


Figura 1.6: Posición de los Joints en las diferentes partes de un cuerpo humano.

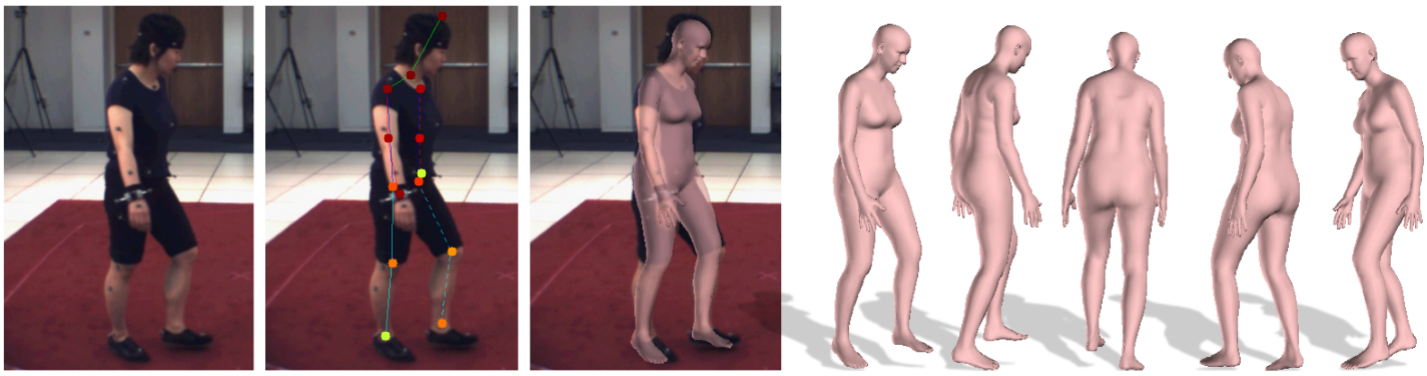


Figura 1.7: De izquierda a derecha: dada una imagen, se usa CNN para predecir las posiciones 2D de los joints (los colores más cálidos denotan alta confianza). Se ajusta un template (modelo 3D) para estimar la pose y el shape. Por último, se muestra el resultado desde varias vistas

En [Romero et al., 2017, Li et al., 2017] se utilizaron los avances nombrados para obtener también la forma, posición de las manos y la expresión de la cara de la persona.

Todo esto permitió que en 2019 se publicara el artículo **Expressive Body Capture: 3D Hands, Face, and Body from a Single Image** [Pavlakos et al., 2019] en el se juntaban todos los avances nombrados anteriormente para generar un modelo 3D de un cuerpo humano realista. En la siguiente imagen se puede ver como en los modelos generados con SMPL no varìa ni la cara ni las manos, en cambio en los generados utilizando SMPL-X sí.

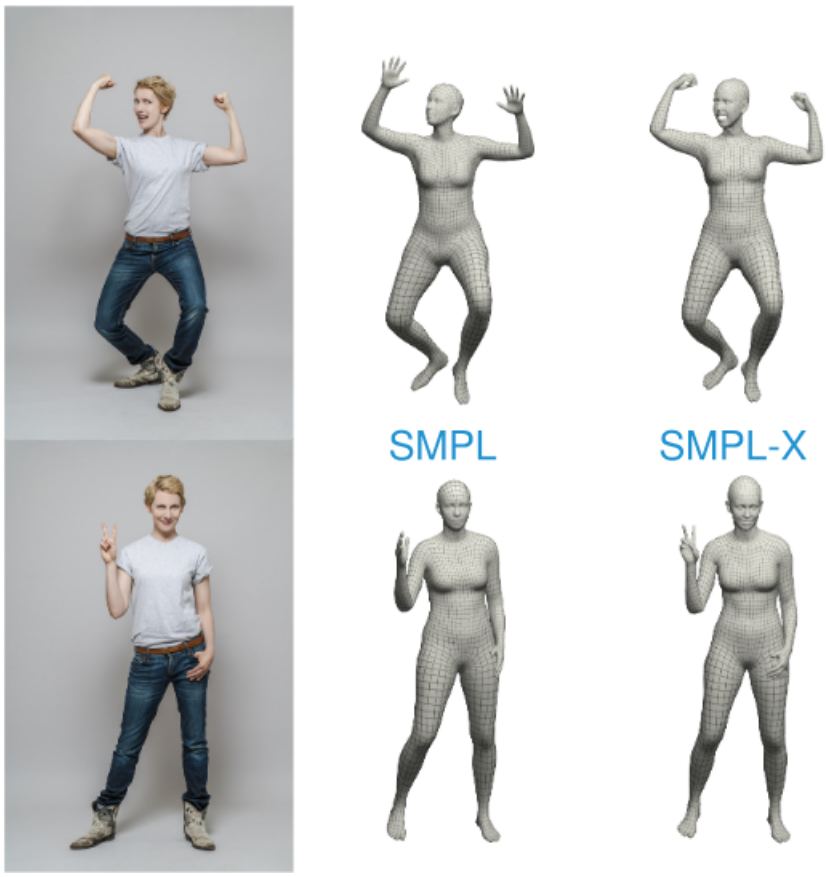


Figura 1.8: Modelos obtenidos a partir de la imagen RGB de la izquierda utilizando SMPL y SMPL-X

Por último, en 2020 apareció STAR para mejorar notablemente a SMPL como modelo para la creación y análisis de cuerpos humanos. La mejora con respecto a los anteriores métodos la consiguen definiendo correctivos a cada joint y al subconjunto de vértices de la nube de puntos que están influenciados por cada movimiento de la articulación. Esto da lugar a deformaciones más realistas. Realizando un entrenamiento sobre el mis-

mo dataset, el modelo STAR generaliza mejor. Por otro lado, en los modelos STAR las deformaciones ya no son pose-dependientes. En otras palabras, si tenemos dos modelos, uno delgado y otro con obesidad y realizan un movimiento, SMPL deformará los dos modelos según el movimiento, es decir según el cambio en la pose. En cambio, STAR tendrá en cuenta la forma, la pose y el BMI (Body Mass Index) para realizar el movimiento.

1.5. Objetivos

Este trabajo de final de máster comparte el mismo objetivo general que el proyecto en el que está enmarcado y es estudiar la evolución del cuerpo humano utilizando tecnologías de visión 3D. Para conseguir esto se plantean los siguientes objetivos parciales:

- **Objetivo 1: Realizar un sistema capaz de obtener los datos médicos y geométricos de las personas examinadas en el proyecto Tech4Diet.** Para ello se han realizado las siguientes tareas:
  - Recopilación a lo largo del tiempo de datos médicos de los pacientes.
  - Realización a lo largo del tiempo de capturas 3D de los pacientes.
  - Automatizar proceso de obtención de las medidas del cuerpo humano utilizando nubes de puntos.
- **Objetivo 2: Realización de un preprocesado en los data para su posterior utilización.**
  - Identificar los valores atípicos dentro de los datasets obtenidos de las diferentes fuentes.
  - Remplazar valores nulos utilizando técnicas de imputación.

- Anonimizar los datos para preservar la privacidad de los pacientes pertenecientes al proyecto Tech4Diet.

- Eliminación de datos duplicados tanto en filas como en columnas.

- **Objetivo 3: Realización de un análisis de los datos.**

  - Predicción de variables utilizando métodos de regresión. Se utilizarán un total de 10 modelos de regresión diferentes.

  - Evaluación de la eficacia de los modelos de regresión con el uso de varias métricas.

  - Reducción de la dimensionalidad y clusterización de los datos para encontrar patrones.

2 Obtención de los datos

En este capítulo se explicarán los diferentes métodos desarrollados para la obtención de los datos asociados a los pacientes evaluados en el proyecto Tech4Diet. Durante el transcurso del 2020, en el proyecto Tech4Diet se ha realizado un estudio dietético-nutricional a 87 voluntarios.

Estos voluntarios han sido divididos en dos grupos. Un grupo control el cual ha pasado el tratamiento nutricional-dietético de forma tradicional y otro grupo, llamado experimental, que además del tratamiento en cada sesión podían ver su progreso en el tratamiento con unas gafas de realidad virtual.

Figura 2.1: Representación de la visualización de los modelos 3D en unas gafas de realidad virtual.

Aunque se realizó esta división en grupos, a todos los participantes en el proyecto se les dijo que deberían pasar por la cabina para obtener un modelo 3D de su cuerpo en cada sesión. Por lo tanto, disponemos de datos 3D de todos los pacientes participantes en el proyecto. En las siguientes secciones se explicarán los diferentes softwares utilizados para obtener los datasets que se analizarán. Primero, se explicará de donde provienen los datos médicos. Luego, detallará la fuente de los datos psicológicos. Por último, se explicará como se ha realizado la obtención de modelos 3D del cuerpo humano.

2.1. Software Tech4Diet

En los últimos años, en el proyecto Tech4Diet se ha implementado en Unity un software que permite el almacenamiento, modificación y eliminación de los datos que son recopilados por el especialista durante las sesiones. Con el uso de la aplicación el especialista ha creado un perfil para cada paciente. Esos perfiles quedan almacenados en una base de datos. A esta tabla se la ha llamado **pacientes** y tiene los siguientes campos:

Tabla 2.1: Columnas y tipo de datos pertenecientes a la tabla **pacientes**

Column	**Dtype**
id	int64
dni	object
nombre	object
apellidos	object
sexo	object
domicilio	object
poblacion	object
numero_ss	object
telefono	int64
email	object
edad	int64
altura	int64

En las siguientes imágenes se puede ver cuál es la pantalla de inicio del software

desarrollado. En la imagen 3.1 se puede observar a la izquierda el menú al que se deberá acceder para añadir un nuevo paciente a la base de datos.

Figura 2.2: Pantalla de inicio del software desarrollado en Tech4Diet.

En la figura 2.3 se pueden observar el formulario necesario para crear un perfil a nuevo paciente. Es necesario rellenar todos los campos.

Figura 2.3: Pantalla donde el especialista crea el perfil del paciente.

Une vez creado el perfil se puede comenzar a pasarle cita al paciente. En cada sesión antes de realizar la captura 3D del cuerpo se debe pasar por una tanita para recopilar datos médicos. Una tanita es una báscula de análisis corporal de la cual se pueden obtener datos como el imc, el peso, el nivel de grasa corporal global y por diferentes partes del cuerpo (pierna derecha, pierna izquierda, brazo derecho, brazo izquierdo y tronco), también se pueden obtener los niveles de musculatura global y por partes o el nivel de grasa visceral entre otros. En este proyecto se ha utilizado la TANITA MC-780MA P 2.5.

Figura 2.4: TANITA MC-780MA P.

Además de estos datos, en cada sesión se le tomará la tensión al paciente y se le medirá el nivel de glucosa, triglicéridos y colesterol en sangre. También se le medirá el perímetro de la zona de la cintura, muñeca y cadera. Todos estos datos también son almacenados por la aplicación diseñada. Después de realizar una captura tendremos un formulario a nuestro alcance para rellenarlo con los datos obtenidos.

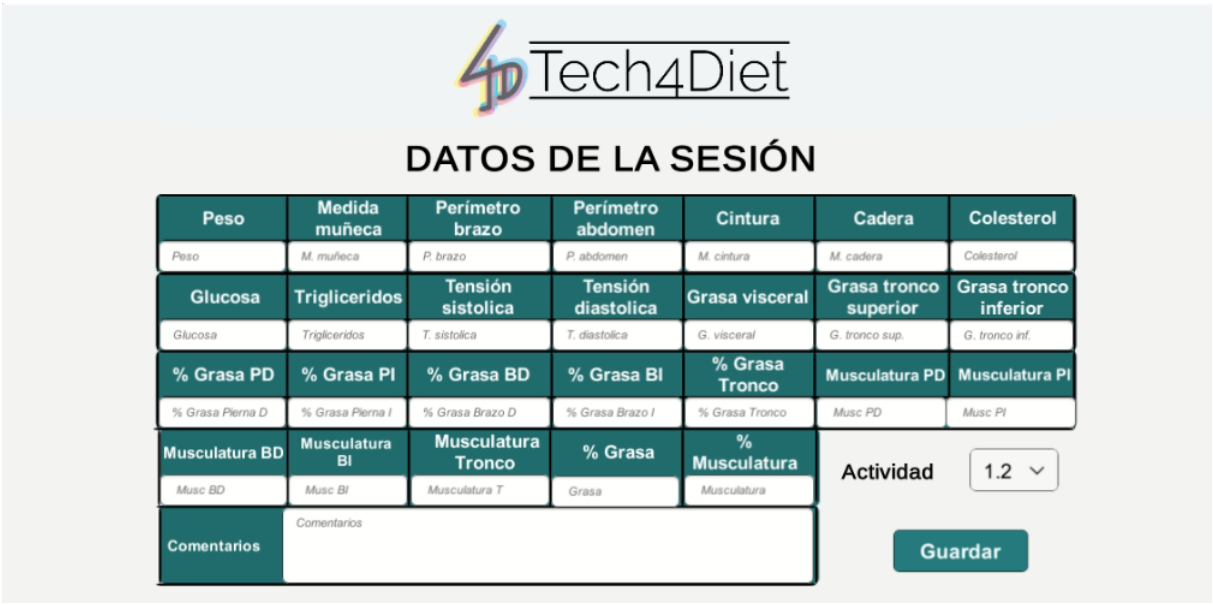


Figura 2.5: Formulario para introducir los diferentes datos recopilados durante una sesión.

Al igual que con los datos personales del paciente, estos datos serán almacenados en la base de datos en la tabla **sesiones**.

Para obtener esas tablas se ha utilizado la herramienta HeidiSQL. Se ha configurado el servidor para que mediante un tunel SSH se pueda acceder a la base de datos externamente. Y con la opción de HeidiSQL de exportar la tabla hemos obtenido un CSV para la tabla paciente y otro para la tabla sesiones.

2.2. Software Cognifit

El software de Cognifit ha sido el que han utilizado los especialistas en el campo de la psicología para medir diferentes habilidades cognitivas del paciente antes y después de realizar el tratamiento nutricional. Para la medición se ha utilizado la Batería de Evaluación Cognitiva General (CAB). El CAB es una herramienta profesional, que nos

permite evaluar una serie de habilidades como: la velocidad de procesamiento, la flexibilidad cognitiva, la percepción visual y auditiva, la memoria contextual, la memoria corto plazo y no verbal, la coordinación ojo-mano, etc.

Todas estas mediciones nombradas anteriormente no se han podido obtener de forma individual debido a problemas con la privacidad de los datos. Aun así si que he podido obtener el permiso para usar el resultado global del test antes y después de realizar el tratamiento nutricional.

2.3. Herramienta Scanserver

Esta es otra de las herramientas desarrolladas en el proyecto Tech4Diet. Esta herramienta es la encargada de realizar diversas tareas como:

- Adquisición de las nubes de puntos. Se obtiene una nube de puntos por cada cámara que realiza la captura.
- Preprocesado de las nubes de puntos. Se aplican una serie de filtros (filtro de la mediana, filtro bilateral y filtro estadístico para eliminar puntos atípicos (Statistical outliner removal filter, SOR)). Además, se aplica un truncado en el eje Z para eliminar los puntos que están más lejanos de la zona de captura deseada.
- Registro de las nubes de puntos, en este proceso se unifican las nubes de puntos en una sola. Para ello se utilizan las matrices de transformación que han sido obtenidas en el calibrado de las cámaras.
- Creación de la malla.
- Proyección de la textura sobre la malla creada.

En la siguiente imagen podemos ver una representación del pipeline que sigue la aplicación en cada captura.

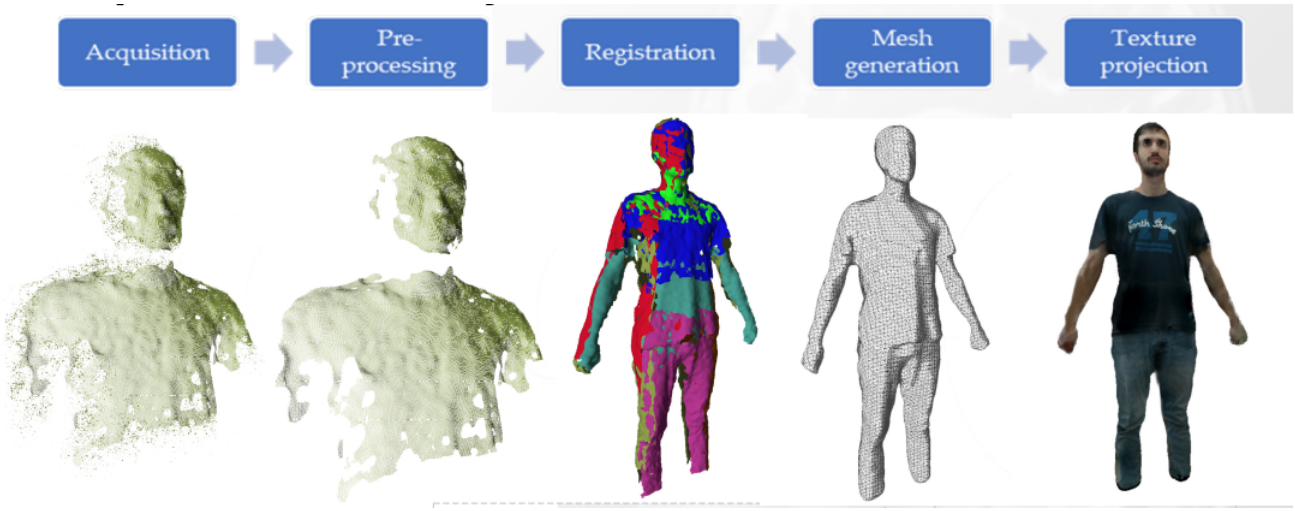


Figura 2.6: Pipeline realizado por la aplicación Scanserver.

Sobre este pipeline, en el momento posterior a la creación de la malla, se han realizado cambios para aplicar un **registro deformable CPD**. Con esto se ha buscado que las nubes de puntos generadas no tuvieran zonas sin puntos. Esto es una tarea muy importante, ya que más adelante para realizar el proceso de obtención de medidas antropométricas si el modelo dispone de zonas vacías las medidas no se podrán obtener o se obtendrán con una fiabilidad de los datos muy baja. Con este registro deformable lo que se ha buscado es ajustar un template a la malla para así rellenar los huecos. En nuestro caso, debido a la estructura de la cabina y la posición de las cámaras las zonas de las axilas y entrepiernas no son capturadas de forma completa. En la imagen 2.8 se muestra una representación de este proceso desarrollado. Para que el ajuste del template sobre el modelo sea más preciso se ha utilizado un template para los hombres y otro para las mujeres.

Figura 2.7: Templates utilizados en el registro deformable. A la izquierda el template para los hombres y a la derecha el template para las mujeres.

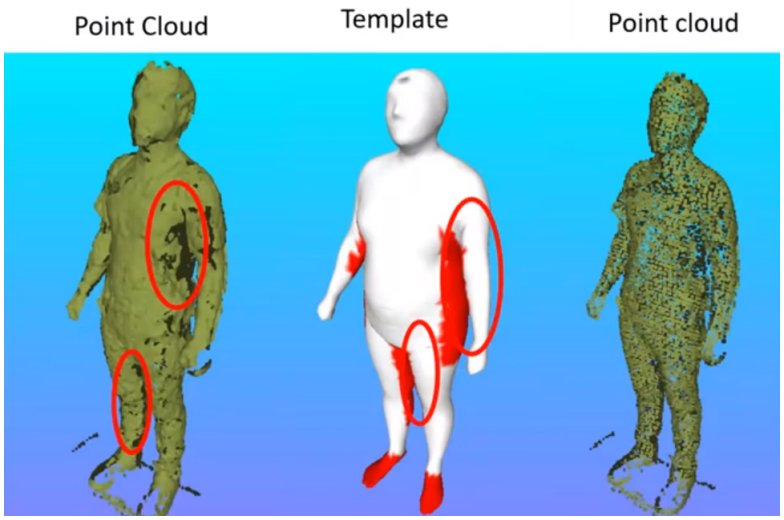


Figura 2.8: Representación del registro deformable aplicado a la malla.

Además, se ha aplicado una reducción en el total de puntos de la malla sin perder la forma de la misma. Para ello se ha utilizado el método desarrollado por Michael Garland en Surface Simplification Using Quadric Error Metrics [Garland and Heckbert, 1997]. En las imágenes 2.11 se puede observar como se contraen los triángulos en la malla y la secuencia de una malla que pasa de 5,804 triángulos a 64. Para aplicar este algoritmo me he apoyado en la implementación realizada por el paquete de Python Open3D [Open3D, 2021].

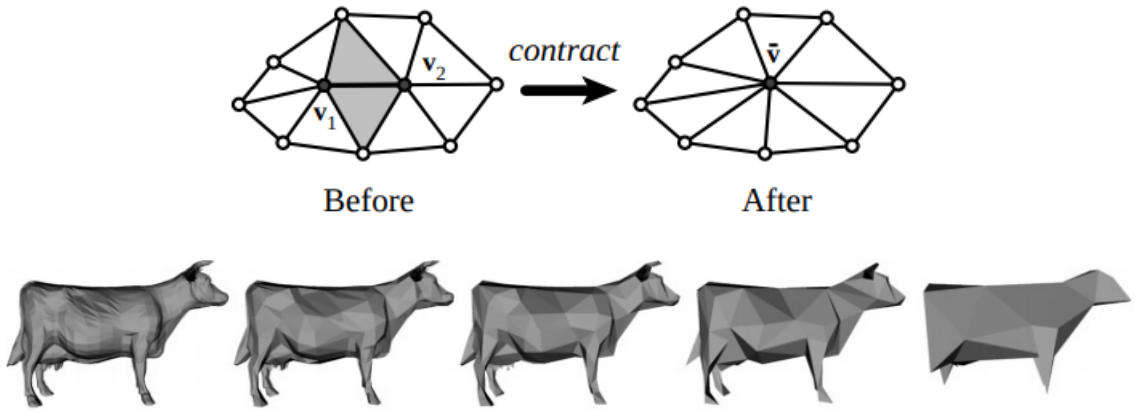


Figura 2.9: Ejemplo de utilización del algoritmo Surface Simplification Using Quadric Error Metrics.

Todo este proceso nos ha permitido obtener un total de 269 modelos de cuerpos humanos y de templates que han sido ajustados utilizando un registro deformable. En las siguientes imágenes se pueden observar los modelos y templates generados para dos pacientes del proyecto. Para conservar el anonimato de las personas se ha eliminado la textura de los modelos 3D.

Figura 2.10: en las dos imágenes se encuentra a la izquierda el template generado y a la derecha el modelo 3D capturado por el sistema.

2.4. Obtención automática de medidas antropométricas

Para la obtención de las mediciones antropométricas de forma automática en los modelos se han utilizado tanto los modelos 3D como los templates asociados a cada uno de los modelos 3D. Para explicar esta parte antes hay que aclarar, que los modelos 3D están compuestos por vértices en el espacio. Cada vértice representa un punto (x, y, z) en el espacio y tiene un id que lo identifica.

Por lo tanto, cada modelo está compuesto por 6500 puntos en el espacio y estos puntos tienen un id asociado. Pero en cada modelo, el orden de los ids es diferente. Esto genera una problemática muy grande a la hora de crear un algoritmo que itere sobre los diferentes modelos y coloque una circunferencia, como se puede ver en la figura 2.12, al rededor de una zona para obtener la medida antropométrica de la zona del cuerpo deseada.

Por el contrario, los templates a los que se le ha aplicado el registro deformable tienen siempre el mismo orden de ids. Esto quiere decir que, si colocamos una circunferencia en la zona de la cintura para un template X y obtenemos los ids de los vértices que corta la circunferencia, para un template Y que tiene una forma diferente al temaplte X (más alto y más ancho por ejemplo) se podrá colocar igualmente la circunferencia en la zona de la cintura, ya que los ids de los vértices que conforman la cintura son los mismos aunque la forma varié.

Por lo tanto, se ha solucionado el problema de la siguiente forma:

- Se han obtenido las diferentes partes a medir de los templates. TemplateWoman para las mujeres y TemplateMan para los hombres.

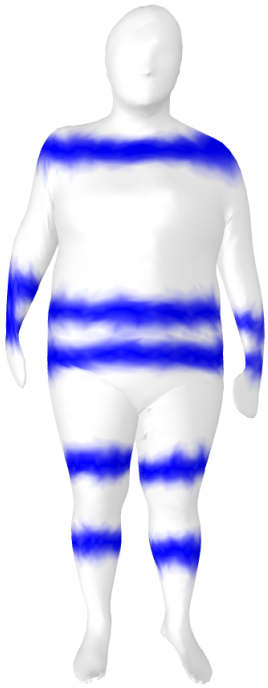


Figura 2.11: Ejemplo de template con las zonas marcadas a medir.

- Luego, se coloca el modelo 3D del cuerpo humano del paciente en la misma posición en el espacio que el template. Para esto se han probado diversos métodos como procrustes e ICP. Pero ha sido el uso de uno de los métodos implementado en la librería de Trimesh de Python el que mejores resultados ha dado. Este método realiza un alineamiento de dos nubes de puntos utilizando los ejes principales de inercia como punto de partida que se refina mediante ICP [Trimesh, 2021].

- Una vez tenemos las dos nubes de puntos alineadas se aplica el algoritmo de KD-Tree implementado dentro de la librería sklearn de Python. Con en el uso de este algoritmo lo que se busca es encontrar en el modelo aquellos puntos que estén más

cercanos a las zonas a medir que están marcadas en el template.

- Con los puntos del modelo que hacen referencia a las zonas a medir ya obtenidos se aplica el algoritmo desarrollado en Multidimensional Measurement of Virtual Human Bodies Acquired with Depth Sensors [Fuster-Guilló et al., 2021]. Este algoritmo se basa en lanzar rayos a lo largo de una circunferencia y los puntos de colisión con el modelo 3D son almacenados para calcular las medidas. En las imágenes siguientes se puede ver un ejemplo de este algoritmo y una visualización de la obtención de medidas para un modelo.

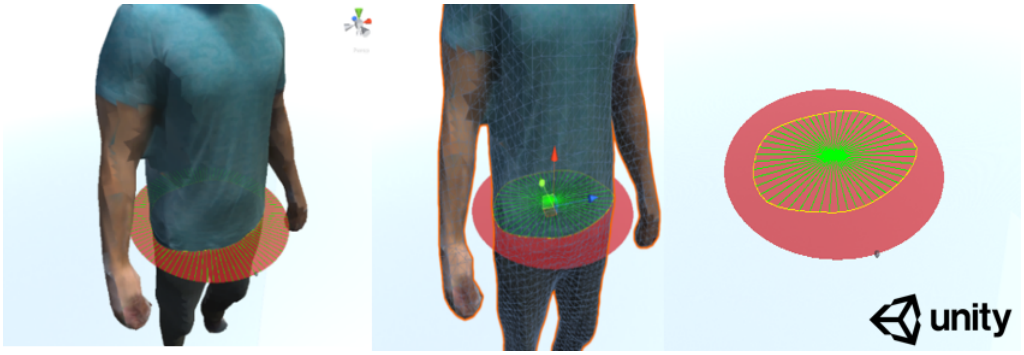


Figura 2.12: Ejemplo de template con las zonas marcadas a medir.

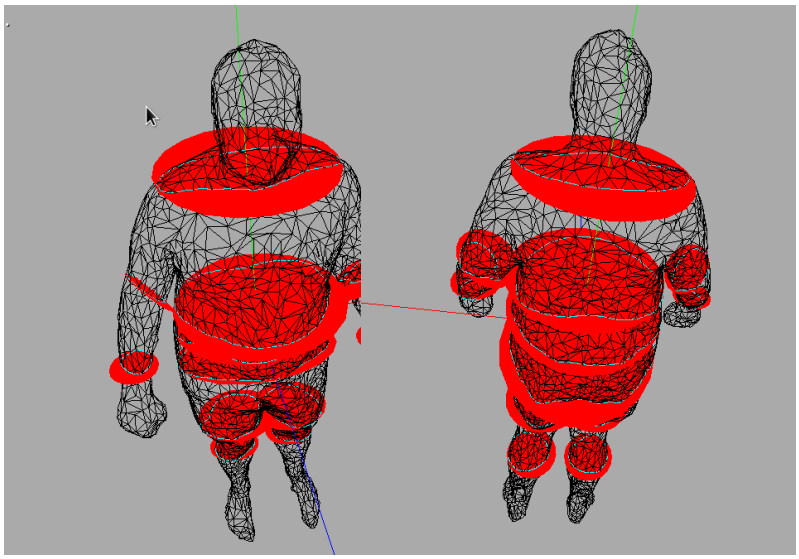


Figura 2.13: Circunferencias situadas en un modelo para obtener las medidas antropométricas (vista delantera y trasera)

2.5. Precisión en la obtención de medidas antropométricas automatizadas

El proceso descrito en el punto anterior no es algo sencillo. El estado actual en este campo ha avanzado mucho en los últimos años como hemos visto en el estado del arte con la aparición de smpl-x o star 1.4. En nuestro caso, al aplicar todo el proceso descrito anteriormente, en algunos casos los templates generados no son lo suficientemente similares a los modelos 3D como para poder realizar medidas antropométricas sobre ellos de forma fiable.

La razón por la que se produce esto, es porque la creación de los template para un mismo paciente se realiza de forma iterativa sobre sus sesiones, es decir, para la primera sesión se calcula el template partiendo de un template con una forma de cuerpo estándar y, para las siguientes sesiones, se utiliza el template de la sesión anterior para ajustarlo al modelo de la sesión actual. Este proceso tiene como problema, que si el primer template generado no se ha ajustado correctamente, al modelo 3D, en los siguientes se propagará ese error o lo aumentará.

Además, si existe una sesión donde la adquisición del modelo 3D no ha sido buena (alguna cámara se ha movido y hay que volver a realizar el calibrado de las cámaras, o el paciente se ha movido) el template para esa sesión no se ajustara correctamente con el método del registro deformable y, por lo tanto se generará un error que se propagará.

Actualmente, en el proyecto Tech4Diet, se está trabajando en la detección y solución de estos errores utilizando técnicas descritas en SMPL-X y STAR. Por todo esto, se ha decidido que para este trabajo de final de máster no se utilizarán las medidas obtenidas de forma automática con el método desarrollado, sino que se utilizarán las medidas antropométricas obtenidas por el especialista en cada una

de las sesiones del tratamiento nutricional-dietético.

3 Tratamiento y preprocesamiento de los datos

En este capítulo se mostrarán las diferentes técnicas utilizadas para ”limpiar” los datos que han sido obtenidos para el análisis de la evolución morfológica del cuerpo humano. Para toda la fase de preproceso se ha utilizado el lenguaje de programación Python junto a las librerías open source de libre uso como Pandas, Scikit-learn o Numpy.

La librería Pandas ha sido utilizada como la base para todo este proceso. Pandas nos ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales. El nombre Pandas deriva del término ”panel data”. [Pandas, 2021]

Lo primero que se ha realizado es la importación de las librerías que se van a utilizar y de los datos obtenidos. En total tenemos tres tablas. Una tabla para los datos obtenidos de la aplicación de Cognifit y dos tablas asociadas a los datos recopilados de los pacientes durante el tratamiento, a estas dos últimas tablas se les denominará tabla paciente y tabla sesión.

En la tabla sesión nos encontramos todos los datos recopilados en las sesiones del tratamiento nutricional-dietética. Las variables columnas del dataframe son las siguientes:

- **cliente**: variable numérica que indica el id del paciente.
- **sesion**: variable numérica que indica la sesión en la que se han recopilado los datos

- **peso**: Peso en kilogramos del paciente en la sesión.
- **imc**: Variable que indica el valor de Índice de Masa Corporal. Se calcula dividiendo el peso en kilogramos por el cuadrado de la altura en metros.
- **p_brazo, p_abdomen, m_cintura, m_cadera y medida_m**: Hacen referencia a las medidas perimetrales del antebrazo, abdomen, cintura, cadera y muñeca.
- **colesterol**: Valor de colesterol en sangre. Se mide en miligramos por decilitros de sangre (mg/dL).
- **glucosa**: Nivel de glucosa en sangre. Obtenidos en miligramos por decilitro (mg/dl).
- **trigliceridos**: Valor de triglicéridos en sangre (mg/dl).
- **t_sistolica, t_diastolica**: almacenan los valores de presión sistólica y diastólica del paciente para una sesión. Se mide en milímetros de mercurio, mmHG.
- **grasa_v:** Valor del índice de grasa visceral que indica la cantidad de grasa que rodean los órganos. Puede tener valores que van desde 1 hasta 59.
- **grasa_sup y grasa_inf** para indicar el porcentaje de grasa en la zona superior del cuerpo y en la zona inferior.
- **icc**: Índice de cintura-cadera. Se calcula dividiendo el valor de la medida de la cintura con el de la cadera.
- **complexion**: Valor numérico obtenido de la división de la altura (cm) por la medida de la muñeca (cm).
- **gcorporal y mcorporal** para indicar el porcentaje global de grasa o musculatura en el paciente en una sesión.

- **coment**: Variable categórica. En este apartado el especialista podía poner algún comentario relevante de la sesión.

- **tmb**: Valor de tasa Metabólica Basal. Calculada de la siguiente forma:

  - Si era una persona mayor de 65 años y hombre: $tmb = peso(kg) * 11,7 + 588$
  - Si era una persona mayor de 65 años y mujer: $tmb = peso(kg) * 9,1 + 659$
  - Si era una persona menor de 65 años y hombre: $tmb = 66{,}773 + peso(kg) * 13{,}751 + 5 * altura(cm) - 6{,}775 * edad$
  - Si era una persona menor de 65 años y hombre: $tmb = 655{,}095 + peso(kg) * 9{,}563 + 1{,}849 * altura - 4{,}675 * edad$

- **get:** Valor que indica el Gasto de Energía Total. Calculado de la siguiente forma $tmb * actividad$

- **actividad:** Tiene tres posibles valores (1.2, 1.4 y 1.6) y indica el valor de actividad física realizada por el paciente en su día a día

  - 1.2 hace referencia a personas sedentarias, en silla de ruedas, etc.
  - 1.4 hace referencia a personas que realizan poca actividad física.
  - 1.6 personas que realizan un trabajo intenso, realizan deporte de forma continua, semi-profesional o profesional.

- **fecha:** Variable de tipo fecha en formato: "día-mes-año hora:minuto:segundos"

- **directorio:** Variable de tipo categórica que indica el directorio donde se encuentra el modelo 3D de la sesión para el paciente (cliente en este caso).

- **niv_grasa_pd, niv_grasa_pi, niv_grasa_bd, niv_grasa_bi, niv_grasa_tronco:**

Hacen referencia al porcentaje de grasa en cada una de las partes del cuerpo (pierna derecha e izquierda, brazo derecho e izquierdo y tronco).

- **niv_musc_pd, niv_musc_pi, niv_musc_bd, niv_musc_bi, niv_musc_tronco:** Hacen referencia al porcentaje de musculatura en cada una de las partes del cuerpo (pierna derecha e izquierda, brazo derecho e izquierdo y tronco).

En el segundo dataframe, paciente, se encuentran todos los datos asociados a los perfiles de los pacientes.

- **id:** Valor numérico único para identificar a cada paciente.
- **dni:** DNI del paciente.
- **nombre:** Nombre del paciente.
- **apellidos:** Apellidos del paciente.
- **sexo:** Sexo del paciente.
- **domicilio:** Domicilio del paciente
- **poblacion:** Población en la que reside el paciente.
- **numero_ss:** Código identificatorio del paciente. Este código es creado usando la primera letra del nombre y los últimos tres números del DNI.
- **telefono:** Teléfono del paciente.
- **email:** Email del paciente.
- **edad:** Edad del paciente.
- **altura:** Altura en centímetros del paciente.

Las columnas DNI, nombre, apellidos, domicilio, población, numero_ss, teléfono y email serán eliminadas para conservar la privacidad y el anonimato de los paciente.

En el tercer dataframe encontraremos los valores obtenidos en los test realizados en la aplicación Cognifit.

- **CÓDIGO:** Código identificatorio del paciente
- **Puntuación CAB PRE:** Puntuación obtenida en el test en la primera sesión del tratamiento.
- **Puntuación CAB POST:** Puntación obtenida en el test en la última sesión.

Además de estas columnas, encontraremos las columnas 'VELOCIDAD PROCESAMIENTO', 'ATENCION DIVIDIDA', 'FLEXIBILIDAD COGNITIVA', 'PERCEPCIÓN VISUAL', 'PERCEPCIÓN AUDITIVA', 'ESTIMACIÓN', 'MEMORIA CONTEXTUAL', 'MONITORIZACIÓN', 'DENOMINACIÓN', 'PLANIFICACIÓN', 'MEMORIA VISUAL A C/P', 'MEMORIA A CORTO PLAZO', 'MEMORIA NO VERBAL', 'ESCANEO VISUAL', 'MEMORIA AUDITIVA A C/P', 'MEMORIA DE TRABAJO', 'ATENCIÓN FOCALIZADA', 'INHIBICIÓN', 'RECONOCIMIENTO', 'PERCEPCIÓN ESPACIAL', 'COORDINACIÓN OJO-MANO', 'TIEMPO DE RESPUESTA' y 'CAMPO VISUAL'. Estas columnas no se han comentado debido a que no se han podido obtener los datos sobre estas variables por temas de privacidad y por lo tanto, se encuentran con valores nulos en el dataframe.

Estas tres tablas tienen todas las columnas en el formato idóneo, así que no ha sido necesaria ninguna conversión de tipo.

3.1. Anonimización de los datos

Esta será la primera tarea de preproceso que se realizará sobre los datos. Como ya se ha dicho se eliminará de la tabla paciente las columnas que tiene información personal del paciente.

```
1 paciente = paciente.drop(columns=['dni','nombre','apellidos','domicilio',
  ↪  'poblacion','numero_ss', 'telefono', 'email'])
```

En el dataframe cognifit se sustituirá la columna 'CÓDIGO' por el id del paciente. Esta no ha sido una tarea del todo sencilla, ya que he encontrado errores a la hora de buscar los códigos. Esto se ha producido debido a que existen casos donde el especialista nutricional ha puesto códigos diferentes a los que se han puesto en la aplicación Cognifit.

Un ejemplo de esto seria el siguiente caso (en este ejemplo no se han utilizado nombres relacionados con el proyecto Tech4Diet): para el nombre María Ángeles con DNI 12345678A el especialista pone M678 mientras que en la aplicación se ha puesto Ma678. Además se han encontrado errores de escritura, siguiendo el ejemplo anterior, sería cuando el especialista escribe M679.

3.2. Eliminación de duplicados, valores nulos y atípicos

En este apartado se han corregido los dataframe para que no dispongan de valores duplicados, nulos o atípicos.

Empezando con la tabla cognifit, podemos encontrar algunas filas donde no se dispone

de datos o donde se encuentra vacío el valor del resultado de alguno de los test. Para estos dos casos se ha procedido a la eliminación de estas filas. Esta será la última tarea de preprocesamiento que se realizará sobre esta tabla.

Sobre la tabla paciente, se han eliminado aquellos perfiles de pacientes que corresponden a pacientes creados para realizar pruebas. En la siguiente tabla 3.1 se pueden observar algunos de estos valores. En esta visualización no se han eliminado ninguna columna para que sea más representativo.

Tabla 3.1: Ejemplo de filas con datos de prueba en la tabla paciente

	id	dni	nombre	apellidos	sexo	domicilio	poblacion	numero_ss	telefono	email	edad	altura
0	63	48...	Nahuel...	1	H	1	1	1	1	asd@asd.com	1	1
1	64	48...	nahuel...	1	H	1	1	1	1	1@1.com	1	1
2	141	48...	1	1	H	1	1	1	1	@1.com	1	1
3	1	12...	JuanMi	C	H	Calle	Alicante	123	12...89	jmi@jmi.com	36	190
4	23	45...	jmi	jmi	H	sanvi	sanvi	12...	65...	DFSD@DFDF.ES	45	123

Para realizar esto se han aplicado varios filtros.

- Se han eliminado aquellas filas donde encontramos que la altura es menor a 50 cm o mayor a 300 cm.
- Se han eliminado aquellas filas donde la edad era menor a 18 años. Este proyecto fue enfocado a personas mayores de edad no a niños.
- Se ha utilizado los códigos de la tabla de sesiones para eliminar aquellas personas que no disponían de una cantidad de sesiones atípica. Con esto nos referimos a perfiles con una, ninguna sesión o muchas sesiones.

Para el último paso de eliminación de valores atípicos, se han utilizado tanto gráficos de caja (boxplot) como de dispersión (scatterplot). Un diagrama de caja es un método dentro de la estadística descriptiva que representa gráficamente grupos de datos numéricos a través de sus cuartiles. También tienen líneas que se extienden verticalmente desde

las cajas, denominadas bigotes, que indican la variabilidad fuera de los cuartiles superior e inferior. Además, se suelen utilizar puntos para representar los valores atípicos. Mientras que, un gráfico de dispersión es un tipo de diagrama matemático que utiliza las coordenadas cartesianas para mostrar los valores de dos variables para un conjunto de datos

Utilizando los paquetes de matplotlib y seaborn podemos visualizar estos tipos de gráfica de forma sencilla. En los siguientes gráficos muestra la cantidad de instancias que encontramos por cada número de sesiones totales de los pacientes.

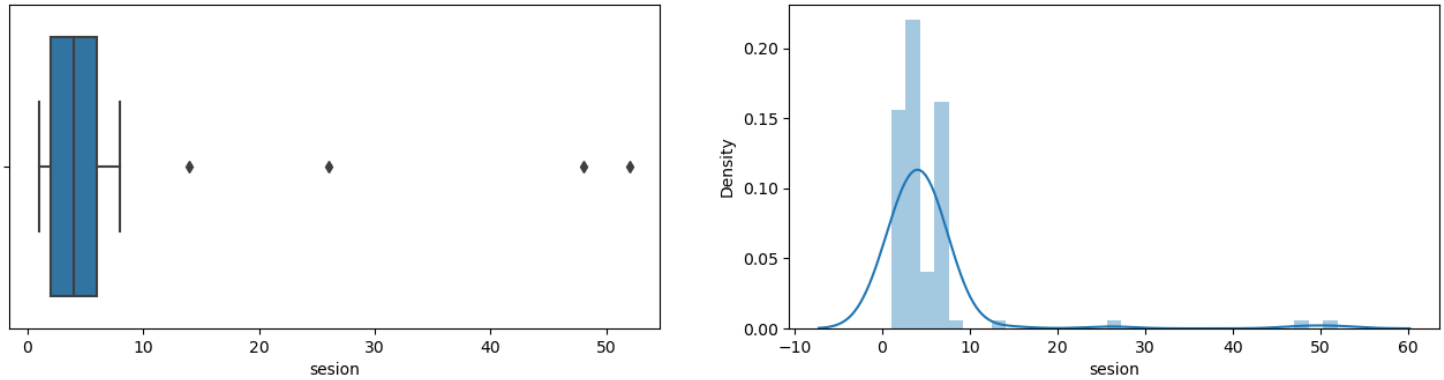


Figura 3.1: Gráfico de caja obtenido a partir del número de sesiones de cada pacientes.

Podemos observar como hay 4 valores atípicos. De estos 4 valores 3 hacen referencia a perfiles de prueba mientras que el punto más cercano al bigote superior si que se corresponde a un perfil de un paciente. Con todo esto el preprocesado de la tabla paciente está acabado.

Ahora solo nos quedaría realizar el procesado de la tabla sesiones. Para esta tabla lo primero que se ha hecho es eliminar aquellos valores que no corresponden a ningún paciente. Usando la tabla de pacientes, ya filtrada, se han eliminado las filas donde el valor de la columna 'cliente' no aparece en ninguna instancia de la columna 'id' de la

tabla paciente. También se han eliminado aquellas sesiones donde el peso es menor a 40. Se ha tomado un umbral bajo para asegurarnos de no eliminar ningún paciente.

Después, se ha eliminado la columna *imc*, ya que es redundante en los datos porque se obtiene del cálculo de dos columnas del mismo dataframe. También se han eliminado las columnas *grasa_sup, grasa_inf, p_brazo, p_abdomen* y *coment*. La razón de su eliminación es que no fueron rellenadas con datos o la cantidad de datos era muy escasa.

Seguidamente, se han generado las gráficas de caja y de dispersión para cada una de las columnas. Dado que son muchas columnas estos gráficos se podrán encontrar en el Anexo 6.1. Observando las gráficas es la columna de **complexión** la que más destaca con 5 valores atípicos por encima de los 150, cuando el resto de valores, de esa misma columna, está entre 1 y 20. Como dijimos anteriormente la complexión es un valor que se calcula dividiendo la altura entre la medida de la muñeca. Volviendo a calcular la complexión obtenemos los siguientes resultados:

Tabla 3.2: Recálculo de la variable complexión para los 5 casos atípicos

Medida muñeca	Altura	Complexión antes	Complexión después
20.40	174	174.0	8.52
16	161	161.0	10,062
18.26	159	159.0	8,707
18.70	177	177.0	9,465
16.15	161	161.0	9,969

Ahora podemos ver como son valores que se encuentran dentro de lo normal, así que se han sustituido. También destacan los valores atípicos dentro de las columnas referidas a los porcentajes de grasa y musculatura por partes. En este caso, los errores vienen dados porque el especialista se olvidó de introducir la coma decimal. En la siguiente tabla 3.3 podemos observar los valores para el cliente número 2 donde se observa la perdida de grasa en la pierna derecha y como de repente en la sesión 11 tiene un valor de 216.

Tabla 3.3: Valores de sesión, y nivel de grasa de la pierna derecha para el cliente número 2

	niv_grasa_pd	sesion	cliente
1	44.85	1	2
2	44.85	2	2
3	24.90	3	2
4	24.80	4	2
5	24.50	6	2
6	23.90	7	2
7	23.00	8	2
8	23.80	9	2
9	23.00	10	2
10	216.00	11	2
11	19.80	12	2

Esto se ha solucionado dividiendo por 10 aquellos valores que se han encontrado como atípicos en las columnas mencionadas. El resultado es el siguiente:

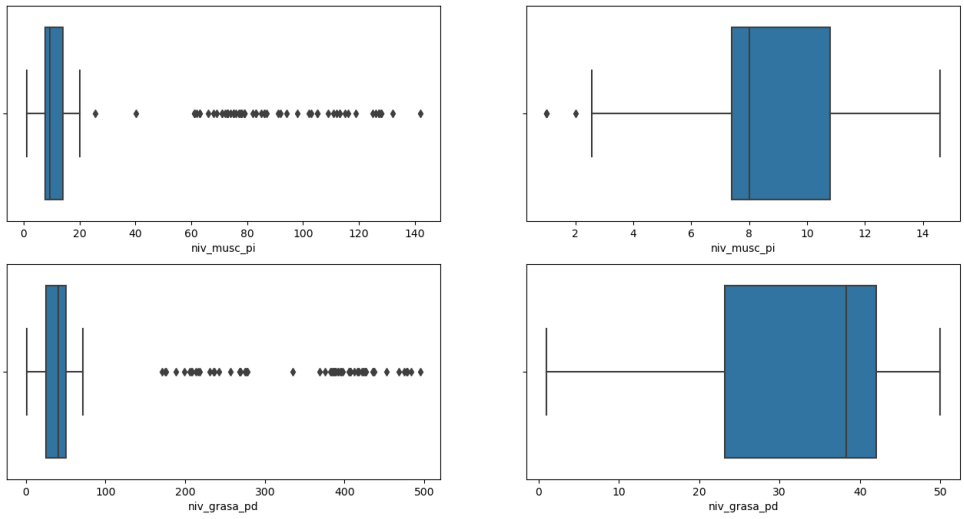


Figura 3.2: Comparativa de los valores de nivel de musculatura de la pierna izquierda y del nivel de grasa de la pierna derecha antes y después de aplicar el filtro para corregir los valores atípicos. A la izquierda se encuentra los boxplot antes de aplicarlo y a la derecha el resultado después de aplicarlo

Los valores atípicos en las variables de porcentaje grasa corporal global y musculatura global se refieren todos al mismo paciente, por lo tanto son valores validos, ya que ese era un paciente el cuál hacía mucho deporte. Se puede saber que hacía mucho deporte, ya que su valor en la **actividad** es de 1.4. Además, en sus datos se puede observar como se ha puesto el siguiente comentario: *"Hace deporte continuo desde hace años, tiene nociones de nutrición y practica el plato de Hardvar."*

Tabla 3.4: Paciente con alto porcentaje de musculatura y bajo nivel de grasa

	mcorporal	actividad	sesion	gcorporal	cliente
1	85.6	1.4	1.0	10.0	24.0
2	85.0	1.4	2.0	10.6	24.0
3	63.2	1.4	5.0	12.4	24.0
4	83.1	1.4	4.0	12.6	24.0
5	82.4	1.4	3.0	13.4	24.0

En el caso del valor atípico en la columna *icc*, es debido también a un error al escribir el dato. El *icc* es calculado con la medida de la cintura y cadera. Y podemos observar como para la sesión 2 en el paciente 86 se produce un error a la hora de escribir el valor de la medida de la cadera.

Tabla 3.5: Tabla que muestra como el paciente, con id 86, que tiene un error en el valor de la medida de la cadera y por lo tanto, un error en el calculo del icc

	m_cintura	sesion	m_cadera	cliente	icc
304	90.0	3.0	112.0	86.0	0.80
305	94.0	1.0	114.5	86.0	0.82
306	91.0	2.0	11.5	86.0	7.91
307	90.5	4.0	111.5	86.0	0.81

Para las columnas t_sistolica y t_distolica se ha encontrado el mismo errores que en las columnas anteriores, error a la hora de introducir los datos (coma decimal mal puesta). Se ha corregido de la misma forma.

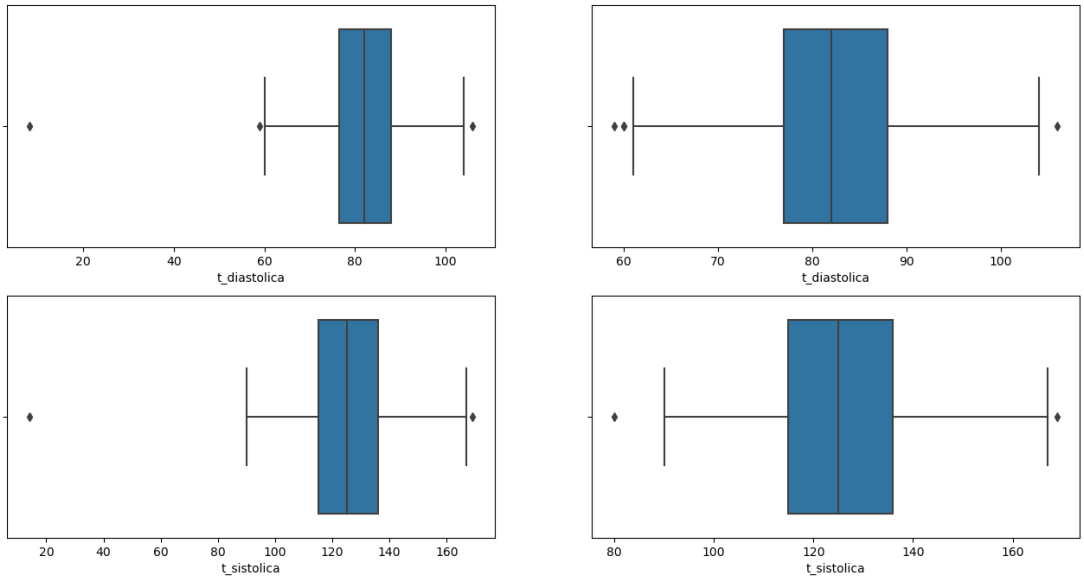


Figura 3.3: Comparativa de los valores de tensión sistólica y tensión diastólica antes y después de aplicar el filtro para corregir los valores atípicos. A la izquierda se encuentras los boxplot antes de aplicarlo y a la derecha el resultado después de aplicarlo

Existe también un valor atípico muy claro en la columna **mcorporal**. Si vemos la tabla del paciente del que proviene este valor nos podemos dar cuenta que es un error de escritura. Este valor se ha sustituido por el correcto 67.2

Tabla 3.6: Tabla que muestra como el paciente 86 tiene un error en la columna mcorporal

	sesion	cliente	mcorporal
1	7.0	3.0	68.9
2	3.0	3.0	68.3
3	5.0	3.0	68.2
4	4.0	3.0	67.8
5	1.0	3.0	672.0
6	2.0	3.0	67.2

Después de todo este proceso descrito, se puede observar en el Anexo 6.5 y 6.6 como los boxplot pasan a tener valores atípicos solo en rangos cercanos a 0. Estos valores están asociados a sesiones en las que no se ha recopilado datos. Por ejemplo, durante las dos primeras sesiones no se han recopilado datos del porcentaje de musculatura o grasa por partes. Estos campos de los que no se había recopilado información fueron introducidos en la aplicación del especialista con unos, ya que la aplicación no permitía valores nulos. Por eso se ha procedido a realizar una imputación de datos, la cual será explicada en el siguiente apartado.

3.3. Imputación de Datos

En estadística la imputación es el proceso de remplazar los valores nulos de un conjunto de datos. Las técnicas de imputación se pueden dividir en dos grandes grupos: las técnicas de imputación simples y las técnicas de imputación múltiple.

En la imputación simple se remplazan los datos faltantes por un único valor. Para cualquier enfoque de este tipo de métodos, se subestiman los errores estándar de las variables en las que faltan datos originalmente al tratar a los datos obtenidos como una muestra completa y no tener en cuenta las consecuencias del método. [XH et al., 2014]. Los métodos que realizan este tipo de imputación asumen que los datos faltantes siguen un patrón MCAR (Missing Completely At Random) [Rosas and Verdejo, 2009]. Una técnica de este tipo de imputación puede ser la imputación con la media no condicional. Este procedimiento preserva el valor medio de la variable, pero los estadísticos que definen la forma de la distribución (varianza, percentiles, sesgo, etc) pueden verse afectados.

La imputación múltiple fue propuesta por Rubin en 1978 [Rubin and Service, 1978]. La imputación múltiple se basa en reemplazar cada valor faltante por un conjunto de m valores obteniéndose así m conjuntos completos de datos, lo que da a lugar a m

estimaciones con sus respectivas varianzas o errores estándar [XH et al., 2014].

En nuestro caso para realizar la imputación se ha utilizado el método KNNImputer de la librería scikit-learn [Scikit-learn, 2021]. Este es un método de imputación simple que realiza la imputación utilizando el valor medio de los N vecinos más cercanos. Para el ejecución de este algoritmo se ha dividido el dataset en 3 partes, una parte con el conjunto de columnas numéricas menos las columnas de sesión y cliente, otra parte con las columnas sesión y cliente, y la última con las variables categóricas. Al terminar todo el proceso de imputación se ha concatenado los tres dataset.

En el Anexo 6.9 y 6.10 se podrán encontrar los gráficos de caja después de aplicar la imputación. Se puede observar como han mejorada notoriamente todas las columnas referidas a la musculatura y grasa (Los valores que pueden parecer atípicos en esas gráficas corresponden a valores reales de pacientes). Con todo esto, el trabajo de preprocesado de los datos está acabado

3.4. Z-Score

Este ha sido el último proceso que se les ha aplicado a los datos para eliminar los valores atípicos que pudiesen quedar. El valor Z-Score es el número de desviaciones estándar de la media de un punto en los datos. Los valores considerados como atípicos serán aquellos en los que el Z-Score sea mayor o menos a 3. La fórmula matemática es la siguiente:

$$Z - score = \frac{x - \mu}{\sigma} \tag{3.1}$$

Donde μ es la media de los datos, σ es la desviación estándar y X son el conjunto de datos.

La aplicación de esta fórmula a nuestros datos se ha hecho utilizando la librería Scipy.

4 Análisis de los datos

En la primera parte de este capítulo se ha realizado un trabajo de análisis exploratorio de los datos que han sido ya preprocesados. Luego, se han detallado los diferentes algoritmos de regresión y clusterización que han sido utilizados para su posterior evaluación. Por último, se analizarán con diferentes métricas los diferentes modelos de aprendizaje automático generados.

4.1. Análisis exploratorio de los datos

En estadística, un análisis exploratorio de los datos (Exploratory Data Analysis, EDA) es un enfoque al análisis de los conjuntos de datos para resumir sus características principales, a menudo con métodos visuales. Se puede utilizar un modelo estadístico o no, pero principalmente el EDA sirve para ver lo que los datos nos pueden decir más allá de la tarea formal de modelización o comprobación de hipótesis [Andrienko and Andrienko, 2005].

Para el análisis de la tabla cognifit primero se ha obtenido de la tabla sesión la primera y última sesión. A esa tabla se le ha añadido una columna llamada **test**. Donde se han añadido el resultado del primer y último test realizado por cada paciente. Un ejemplo visual de lo descrito se puede ver en la tabla 4.1

Tabla 4.1: Ejemplo de tabla creada con los valores de los test de la tabla Cognifit

	cliente	peso	test
0	2.0	109.4	492
1	2.0	94.7	616.0
2	3.0	108.4	471
3	3.0	95.4	629.0

Ya en esta tabla podemos observar como puede llegar a existir una correlación entre el peso de los pacientes y el resultado en los test cognitivos. Obteniendo los valores de correlación se ha observado que no era como pensaba y no hay una correlación fuerte entre las variables test y peso. Las variables que más correlacionadas están con los resultados de los test son las variables que hacen referencia a la musculatura por partes del cuerpo, la musculatura corporal global y la altura.

Tabla 4.2: Variables más correlacionadas con los resultados del test Cognifit

	test
mcorporal	0.388597
altura	0.275380
niv_musc_pd	0.197113
niv_musc_bd	0.188000
niv_musc_pi	0.178550

También se han obtenido las variables más correlacionadas sobre el conjunto total de sesiones. La tabla completa se puede ver en el Anexo 6.2. Obviamente las correlaciones más altas se encuentran en aquellas variables que tratan de lo mismo. Por ejemplo las variables de grasa corporal global con el porcentaje de grasa por partes del cuerpo (lo mismo con la musculatura).

En las medidas antropométricas encontramos que las variables más correlacionadas son las que se muestran en las siguientes tablas.

Tabla 4.3: Variables más correlacionas con las variables antropométricas

	medida_m		m_cadera
peso	0.693074	niv_grasa_tronco	0.681677
niv_musc_pi	0.687441	gcorporal	0.680430
tmb	0.682580	peso	0.630504
niv_musc_pd	0.663375	m_cintura	0.529877
m_cintura	0.651116	niv_grasa_pi	0.401291
grasa_v	0.650647	niv_grasa_bd	0.399605

	m_cintura
peso	0.845360
grasa_v	0.826568
icc	0.740288
tmb	0.687541
medida_m	0.651116
niv_musc_pi	0.600562

4.2. Regresión

La regresión busca estimar o predecir, para cada individuo el valor numérico de alguna variable para ese mismo individuo [Provost and Fawcett, 2013]. En el caso de los modelos donde se predice un valor numérico, se utiliza una medida de precisión para evaluar le eficacia del modelo. Sin embargo, hay diferentes formas de medir la precisión, cada una con su propio matiz. Para medir los puntos fuertes y débiles de un modelo resulta problemática usar solo una métrica. Para los modelos de regresión la métricas más utilizadas son:

- Mean absolute error (MAE)
- Mean squared error (MSE)
- Root mean squared error (RMSE)

- Root mean squared logarithmic error (RMSLE)
- Mean percentage error (MPE)
- Mean absolute percentage error (MAPE)
- R2

En este trabajo se utilizaran el MAE, MSE, RMSE y R2 para medir la calidad de los diferentes modelos que se probarán. Para modelos donde el resultado es un valor numérico el RMSE es la métrica más común. Esta métrica es una función de los residuos del modelo, que son los valores obtenidos menos las predicciones del modelo. El error medio cuadrático (MSE) se calcula elevando al cuadrado los residuos, sumándolos y dividiéndolos por el número de muestras. Y el RMSE se calcula realizando la raiz cuadrada de MSE. El valor se suele interpretar como la distancia entre los valores observados y los predichos.

La métrica R^2 se puede interpretar como la proporción de la información en los datos que es explicada por el modelo. Por ejemplo, un valor de R2 de 0.8 quiere decir que el modelo puede explicar el 80 % de la variación del resultado. Existen varias fórmulas para calcularlo [Kvålseth, 1985]. La forma más sencilla es encontrando el coeficiente de correlación entre los valores observados y predichos (R) y se eleva al cuadrado (R^2)

En el MAE encontramos que todas las diferencias individuales se ponderan por igual. Se calcula como un promedio de diferencias absolutas entre los valores observados y los predichos. Al realizar la diferencia absoluta el MAE es menos sensible a los valores atípicos que el MSE.

Tabla 4.4: Formulas matemáticas para el calculo del MSE, RMSE, MAE y R^2

Mean squared error	$\text{MSE} = (\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2$
Root mean squared error	$\text{RMSE} = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2}$
Mean absolut error	$\text{MAE} = (\frac{1}{n})\sum_{i=1}^{n}\lvert y_i - x_i\rvert$
R squared	$R^2 = \frac{\sigma_{XY}^2}{\sigma_X^2 \sigma_Y^2}$

En las fórmulas anteriores n es el numero de muestras totales, σ_{XY} es la covarianza de (X,Y), σ_X^2 es la varianza de la variable X, σ_Y^2 es la varianza de la variable Y, y_i son los valores observados y x_i son los valores predichos.

Hoy en día, podemos encontrar todas estas fórmulas ya implementadas dentro de la librería de scikit-learn. También disponemos en la misma librería de la implementación de todos los modelos de regresión aplicados en este trabajo.

El principal objetivo en esta parte del proyecto es ser capaces de predecir los valores de Colesterol y Glucosa en los individuos. Para ello, se ha dividido el dataset de sesiones en cuatro conjuntos de datos diferentes. La razón de la división del dataset viene dada por fuente de obtención de los datos.

En el primer conjunto encontraremos aquellas medidas que se pueden llegar a obtener utilizando tecnologías 3D, es decir las medidas antropométricas (medida de la muñeca, medida de la cintura, medida de la cadera y la altura). En el segundo conjunto encontraremos las variables obtenidas por sistemas médicos, concretamente estarán las variables asociadas a los porcentajes de grasa del individuo

En el tercer conjunto, al igual que en el segundo, se encontraran las variables médicas

asociadas a los porcentajes de musculatura del individuo.

Por último, en el cuarto conjunto de datos encontraremos las variables más correlacionadas con cada unas de las variables a predecir.

Con respecto a los modelos que posteriormente se van a evaluar. Durante el proceso de regresión se han probado diferentes modelos de regresión. Estos modelos son: Gradient Boosting Regressor (GBR), Random Forest Regressor (RFR), Extra Trees Regressor (ETR), Ada Boost Regressor (ABR), Linear SVR (LSVR), Decision Tree Regressor (DTR), Huber Regress (huber), ElasticNet (EN), KNeighbors Regressor (KNN), Passive Aggressive Regressor (PAR)

Con la función de sklearn train_test_split, se ha dividido cada uno de los conjuntos de datos en un tres conjuntos (uno para el entrenamiento y otro para los test). Para cada uno de los casos mostrados a continuación.

4.2.1. Interpretación y evaluación de los algoritmos de regresión

Para cada una de las variables a predecir (glucosa y colesterol) se han generado 4 conjuntos de datos diferentes. Un conjunto con las variables antropométricas, otro con las variables referidas al porcentaje de grasa en el cuerpo, otro con las variables referidas al porcentaje de musculatura en el cuerpo y un último conjunto con todas las variables.

En el conjunto de las medidas antropométricas encontraremos las variables de la medida de la cintura, la medida de la cadera, la medida de la muñeca y la altura del paciente. En el conjunto de datos con las variables de musculatura encontraremos aquellas variables que representan el nivel de porcentaje de musculatura por las diferentes partes del cuerpo y la global (*niv_musc_tronco*, *niv_musc_pd*, *niv_musc_pi*, *niv_musc_bd*, *niv_musc_bi*, *niv_musc_tronco* y *mcorporal*.). En el conjunto de datos con las variables de grasa encontraremos aquellas variables que representan el nivel de porcentaje de

grasa por las diferentes partes del cuerpo y la global *niv_grasa_tronco*, *niv_grasa_pd*, *niv_grasa_pi*, *niv_grasa_bd*, *niv_grasa_bi*, *niv_grasa_tronco* y *gcorporal*. .

Predicción de la variable Colesterol

En las tablas 6.3, que se encuentran en los anexos, se encuentran los resultados de las métricas para cada uno de los modelos base. Se puede ver como el modelo creado con el algoritmo *Passive Aggressive Regressor* ha sido el que peores resultados ha dado en todas las métricas. Con respecto a los datos de entrenamiento el modelo creado con *Gradient Boosting Regressor* es el que mejores resultados han dado en todas las métricas. En los datos de test es el modelo de *Extra Trees Regressors* el que mejores resultados ha dado.

Sobre los dos mejores modelos, GBR y ETR, se ha realizado un GridSearch para optimizar los hiperparámeteros. Aparte de esto, también se ha realizado validación cruzada.

Los hiperparámetros optimizados en cada uno de estos modelos han sido los siguientes. Para GBR se han optimizado el número de etapas de *boosting* a realizar, el *lerning rate*, la fracción de muestras que se utilizará para ajustar los *learners* y la máxima profundidad de los estimadores de regresión individuales.

Para ETR se han optimizado el número total árboles, la función para medir la calidad de la división de las ramas, la máxima profundidad de los arboles y el número de *features* a considerar en cada división.

Podemos observar en las siguientes tablas una comparativa de los valores de las métricas con los modelos base y los modelos con los parámetros optimizados. Sobre el conjunto de entrenamiento los dos modelos optimizados obtienen mejores resultados. Mientras que, para el conjunto de test solo es en el modelo creado con ETR que se puede apreciar una mejora con respecto a su modelo base

Tabla 4.5: Métricas obtenidas antes y después de realizar la optimización de los hiperparámetros con los algoritmos Gradiend Boosting Regressor y Extra Trees Regressor para el conjunto de datos de medidas antropométricas

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR Before	12.591742	281.654799	16.782574	0.445668
GBR After	7.672275	103.155092	10.156530	0.796978

	MAE Test	MSE Test	RMSE Test	R_Squared Test
GBR Before	16.762584	549.06747	23.432189	0.097766
GBR After	16.296504	517.38275	22.746049	0.149830

	MAE Train	MSE Train	RMSE Train	R_Squared Train
ETR After	16.002991	524.610990	22.904388	0.137953
ETR Before	8.141547	115.340175	10.739654	0.772996

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR Before	16.972546	546.769806	23.383109	0.101541
ETR After	10.525107	199.804603	14.135226	0.606760

Utilizando el segundo conjunto de datos, conjunto de datos de musculatura, encontramos que los mejores modelos base obtenidos son los mismos que con las medidas antropométricas (Grandien Boosting y Extra Trees). Esto se puede ver en las tablas 6.3. En este caso solo se ha optimizado los hiperparámetros del modelo ETR. El algoritmo de *Passive Aggressive Regressor* sigue siendo el que peores resultados obtiene con mucha diferencia.

En este caso, los resultados no han sido tan favorables como en el anterior. No se aprecian grandes mejoras con respecto al modelo base.

Tabla 4.6: Métricas obtenidas antes y después de realizar la optimización de los hiperparámetros con el algoritmo Extra Trees Regressor para el conjunto de datos de musculatura

	MAE Train	MSE Train	RMSE Train	R_Squared Train
ETR Before	12.238912	281.392160	16.774748	0.446185
ETR After	11.377532	259.870318	16.120494	0.488543

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR Before	16.963503	530.863675	23.040479	0.127678
ETR After	16.466426	525.470963	22.923153	0.136540

Sobre el conjunto de datos que utiliza las variables de grasa corporal se vuelen a repetir los modelos base que mejores resultados dan para el conjunto de datos de entrenamiento y de test. Lo que destaca en este caso es que para el conjunto de test, el algoritmo ETR obtiene perores resultados a la hora de predecir la variable colesterol. En general, todos los modelos obtienen peores resultados.

En este caso, el modelo optimizado ha obtenido resultados buenos sobre el conjunto de entrenamiento pero no sobre el conjunto de test. Esto se puede deber a que esté ocurriendo overfitting. En este algoritmo, esto puede ocurrir cuando la profundidad de los arboles es muy alta. Para este caso el modelo optimizado se ha generado con una profundidad máxima de 16. En los dos modelos anteriores optimizados la profundidad máxima ha sido de 8.

Tabla 4.7: Métricas obtenidas antes y después de realizar la optimización de los hiperparámetros con el algoritmo Extra Trees Regressor para el conjunto de datos de grasa corporal (conjunto de entrenamiento)

	MAE Train	MSE Train	RMSE Train	R_Squared Train
ETR Before	13.469433	327.965560	18.109819	0.354523
ETR After	6.787569	80.094173	8.949535	0.842365

Tabla 4.8: Métricas obtenidas antes y después de realizar la optimización de los hiperparámetros con el algoritmo Extra Trees Regressor para el conjunto de datos de grasa corporal (conjunto de test)

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR After	18.265457	608.267984	24.663090	0.000487
ETR Before	18.271084	603.948783	24.575369	0.007584

Sobre el conjunto total de variables los resultados han sido distintos. En este caso nos encontramos que para el conjunto de test el mejor modelo es el que ha sido creado con el algoritmo de *Random Forest Regressor*

Para RFR los hiperparámetros optimizados han sido el número de árboles total, el número de *features* a considerar en cada división, la profundidad máxima de los árboles, el número mínimo de muestras requerido para cada división en los nodos internos y el número mínimo de muestras requerido para estar en un nodo hoja.

Los resultados se pueden ver en las tablas que se muestran a continuación. Se observan mejoras para todas las métricas aunque no son significativas.

Tabla 4.9: Métricas obtenidas antes y después de realizar la optimización de los hiperparámetros con el algoritmo Random Forest Regressor para el conjunto de datos total

	MAE Train	MSE Train	RMSE Train	R_Squared Train
RFR After	17.358286	576.286547	24.005969	0.053039
RFR Before	10.023204	168.190656	12.968834	0.668980

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR Before	15.804693	496.637213	22.285359	0.183920
RFR After	14.459292	390.818436	19.769128	0.230821

Predicción de la variable Glucosa

Ahora, vamos a pasar a evaluar los modelos sobre la variable glucosa. Utilizando las variables antropométricas encontramos que el modelo que ha obtenido los mejores valores sobre el conjunto de datos de test ha sido ETR y sobre el conjunto de entrenamiento GBR (tabla 6.3). Sin embargo, en este caso se ha decidido utilizar el algoritmo *Ada Boost Regressor*. La razón de esto es que, este algoritmo es el segundo con mejores resultados en los test y el 4 en los entrenamientos.

Los hiperparámetros optimizados han sido el estimador base, donde se han probado *DecisionTreeRegressor(), KNeighborsRegressor(), LinearRegression()* y ninguno, se han utilizado diferentes valores para el número de estimadores y de learning rate también.

Tabla 4.10: Métricas obtenidas para el conjunto de test y entrenamiento sobre el modelo ABR antes y después de optimizarlo.

	MAE Train	MSE Train	RMSE Train	R_Squared Train
ABR Before	13.469433	327.965560	18.109819	0.354523
ABR After	6.787569	80.094173	8.949535	0.842365

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ABR After	18.265457	608.267984	24.663090	0.000487
ABR Before	18.271084	603.948783	24.575369	0.007584

Aunque en el conjunto de entrenamiento si que mejoran muchísimo las métricas obtenidas, en el de test no lo hace. Haciendo uso de las variables de musculatura el mejor modelo obtenido sigue siendo ETR para el conjunto de test. Sin embargo, para este apartado tampoco se ha probado, ya que también se puede observar como *Random Forest Regressor* obtiene una puntuación muy parecida en los datos de test y sobre lso datos de entrenamiento obtiene mejores resultados. El resultado de su optimización ha sido el siguiente.

Tabla 4.11: Métricas obtenidas para el conjunto de test y entrenamiento sobre el modelo RFR antes y despues de optimizarlo.

	MAE Train	MSE Train	RMSE Train	R_Squared Train
RFR After	9.278416	159.338809	12.622948	0.031689
RFR Before	7.849367	107.742297	10.379899	0.330691

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR Before	9.149787	154.528772	12.430960	0.060920
RFR After	8.129498	97.753703	9.887047	0.392741

Aunque es un valor bajo de R2, podemos observar como el obtenido sobre el conjunto de test mejora muchísimo el valor obtenido en el modelo base. La tabla mostrada a continuación muestra la salida de algunos de los valores reales y los predichos por este modelo optimizado.

Tabla 4.12: Valores reales y predichos por el modelo de Random Forest Regressor con los hiperparametros optimizados y utilizando las variables de musculatura

Valor Real	**Valor Predicho**
77.0	80.564384
84.8	82.262963
83.8	80.728571
76.6	86.925000
81.4	85.844444
70.4	74.400000
83.2	85.717647
69.0	73.067647
87.0	74.910000
76.6	78.444444

Podemos ver en la tabla 6.3 como RFR es el mejor modelo para conjunto de test. En este caso el modelo optimizado no ha dado buenos resultados.

Tabla 4.13: Métricas obtenidas para el conjunto de test y entrenamiento sobre el modelo RFR antes y despues de optimizarlo.

	MAE Train	MSE Train	RMSE Train	R_Squared Train
RFR After	9.056061	139.733118	11.820876	0.131960
RFR Before	8.167804	108.698428	10.425854	0.324751

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR Before	9.610032	165.465401	12.863336	-0.005542
RFR After	9.327595	158.678975	12.596784	0.035699

Al igual que en los dos últimos casos, haciendo uso de todas las variables, para predecir la variable glucosa, el mejor modelo base es el modelo de RFR. En este caso se puede observar en la tabla 4.2.1 como para el conjunto de entrenamiento no se produce casi ninguna mejora. Para el conjunto de test, sí que mejora aunque el valor de R2 sigue siendo bajo.

Tabla 4.14: Métricas obtenidas para el conjunto de test y entrenamiento sobre el modelo RFR antes y despues de optimizarlo.

	MAE Train	MSE Train	RMSE Train	R_Squared Train
RFR After	5.943299	62.620049	7.913283	0.610996
RFR Before	6.022263	62.200921	7.886756	0.613600

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR After	8.757226	146.230242	12.092570	0.111351
RFR Before	8.273479	129.229875	11.367932	0.214663

4.3. Clustering

Las técnicas de clustering se aplican cuando no hay una clase a predecir, pero las instancias deben dividirse en grupos naturales [Witten et al., 2011]. Estos grupos refle-

jan, presumiblemente, alguna característica que hace que las instancias del dominio se parezcan más entre sí que con el resto de instancias.

El algoritmo más clásico en el clustering es el *k-means*. En primer lugar, se especifica cuantos clusters (grupos) se buscan, k. A continuación, se eligen k puntos al azar como centros de los clusters. Todas las instancias se asignan a su centro más cercano según la distancia euclídea. Luego, se calcula el centroide, o la media, de las instancias de cada cluster. Estos centroides se toman como nuevos valores centrales para sus respectivos clusters. Por último, se repite todo el proceso con los nuevos centros. Sobre este proceso se itera hasta que los centros de los clusters se estabilizan.

Existen otras técnicas de clustering como Agglomerative Clustering, el cual forma parte dentro de una clase de algoritmos de cluster denominados clustering jerárquico (Hierarchical clustering, HCA). Otro algoritmo es DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Este algoritmo está diseñado para encontrar clusters de forma arbitraria [Ester et al.,].

En este trabajo para la clusterización de los datos se ha utilizado el método de K-means. Además, en este apartado se han utilizado la técnica de Análisis de Componentes principales (PCA) para reducir la dimensionalidad de los datos.

4.3.1. Interpretación y evaluación de los algoritmos de clustering

En esta sección se interpretarán los resultados obtenidos de aplicar clustering a los datos del proyecto Tech4Diet

Para aplicar clustering a los datos de este proyecto, primero se ha aplicado PCA para reducir la dimensionalidad. Para ello, se ha utilizado la clase PCA de sklearn.decomposition. Por defecto PCA solo centra los valores no los escala. Para solucionar esto, se ha aplicado

la función **StandardScaler** de sklearn.

En la siguiente gráfica se puede observar como el primer componente explica un 43 % de la varianza observada en los datos y el segundo un 19 %.

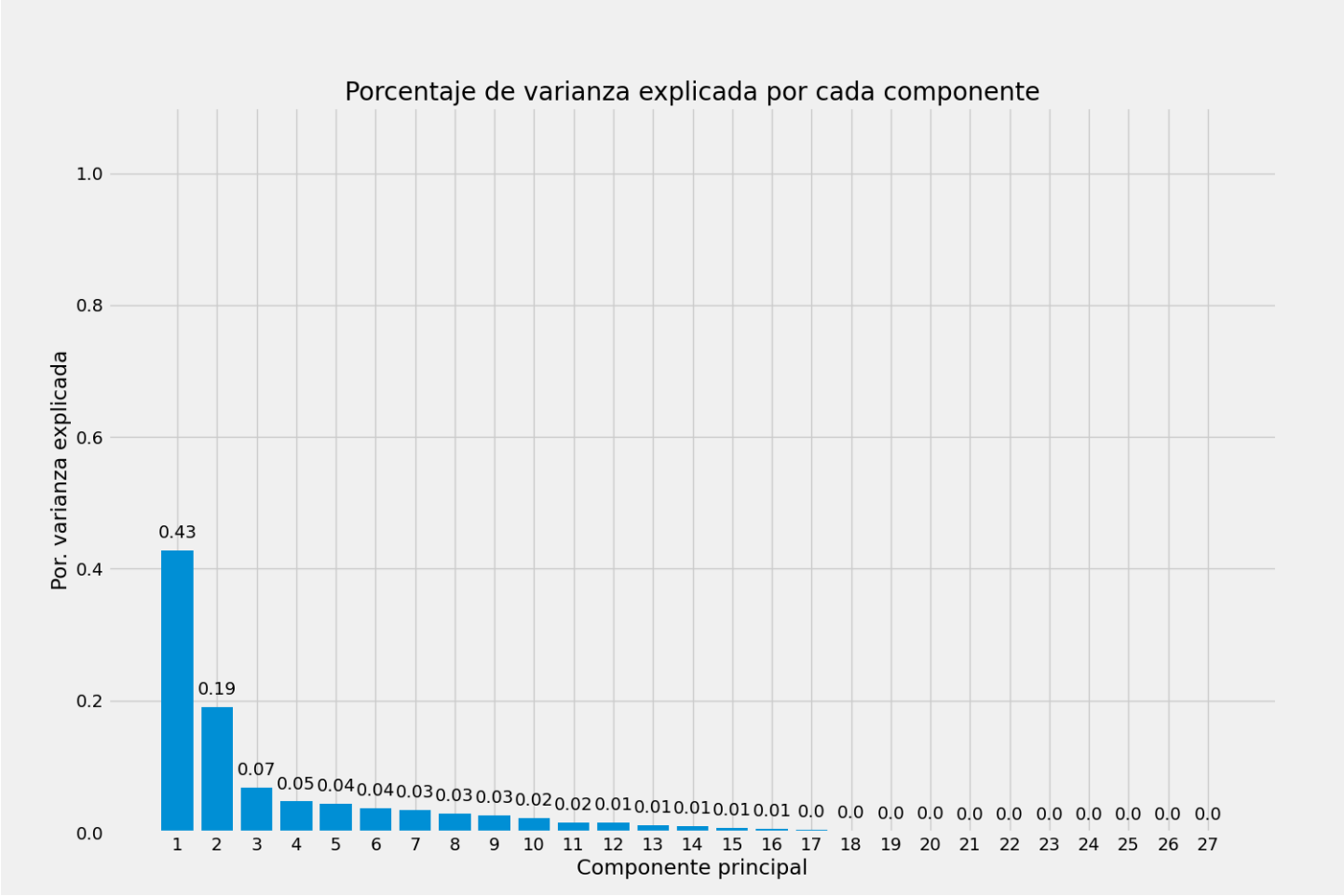


Figura 4.1: Porcentaje de varianza explicada por cada componente para los datos de las sesiones

Con tan solo dos variables estaríamos explicando el 62 % de la varianza observada

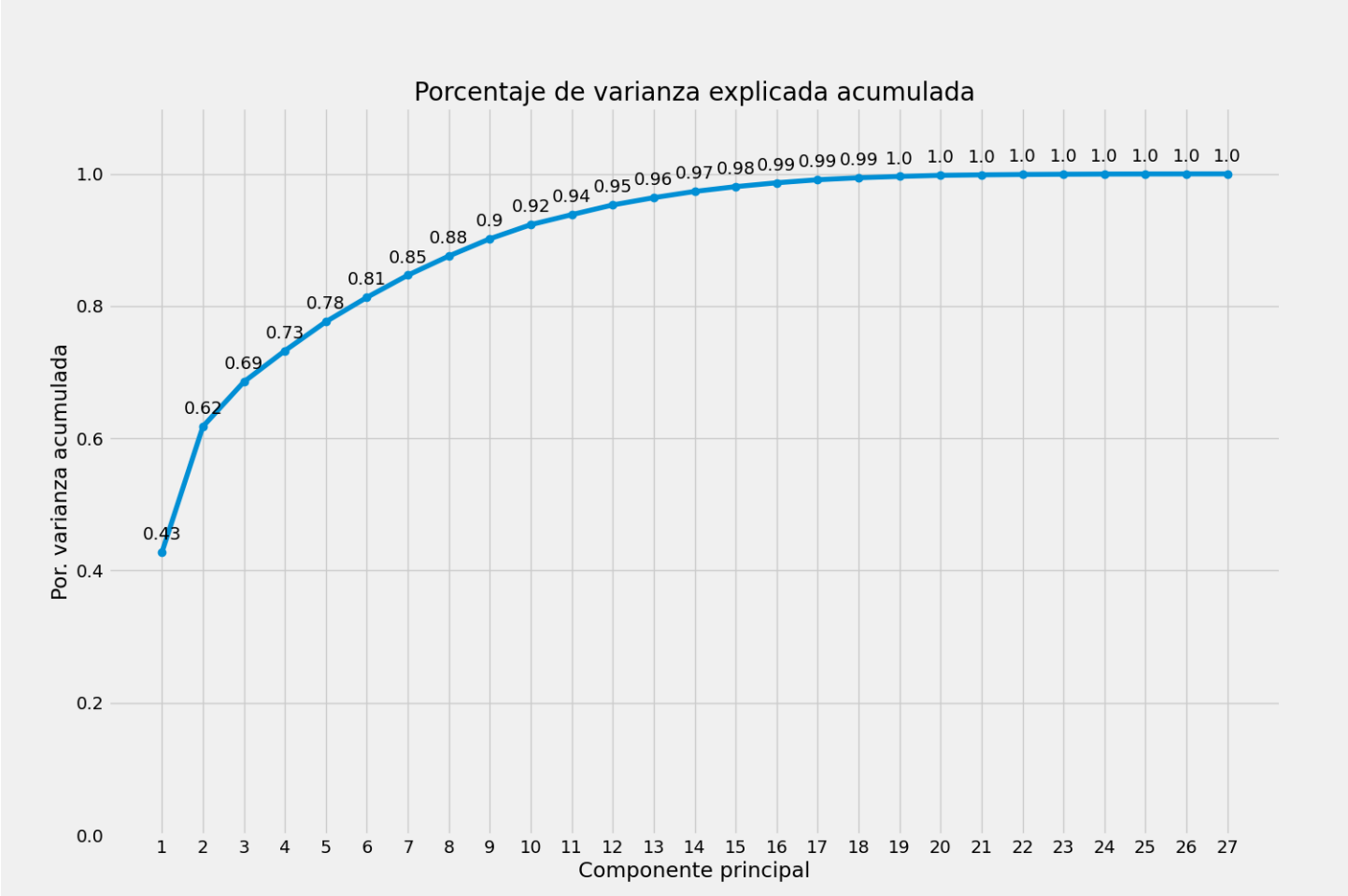


Figura 4.2: Porcentaje de varianza explicada acumulada para los datos de las sesiones

Se ha utilizado, por lo tanto, dos componentes principales.

Partitioning Clustering

Estos tipos de algoritmos requieren que el usuario especifique el número de clusters. El algoritmo más representativo de este tipo es K-Means, que es el que se aplicará en este trabajo.

El número de clusters no es algo que se deba poner al azar. El método del codo es el

más utilizado, este método se basa en ejecutar K-Means para un rango de valores de K clusters e identificar aquel valor a partir del cual la reducción de la *inertia* deja de ser sustancial.

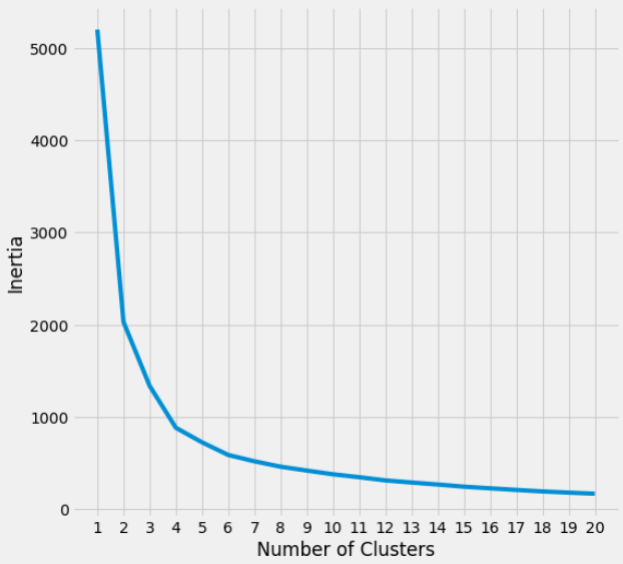


Figura 4.3: Método del codo sobre los datos con dimensión reducida.

Determinar el punto de codo en la curva no siempre es sencillo. Si se tienen problemas para elegir el punto del codo de la curva se puede utilizar un paquete de Python, kneed, para identificar el punto del codo mediante programación

```
from kneed import KneeLocator
kl = KneeLocator(range(1, 21), sse, curve="convex", direction="decreasing")

kl.elbow
# 4
```

El método de la silueta (silhouette), tambien es utilizado para obtener el número óptimo de clusters.

En la figura 4.4 se pueden observar lso diferentes clusters creados.

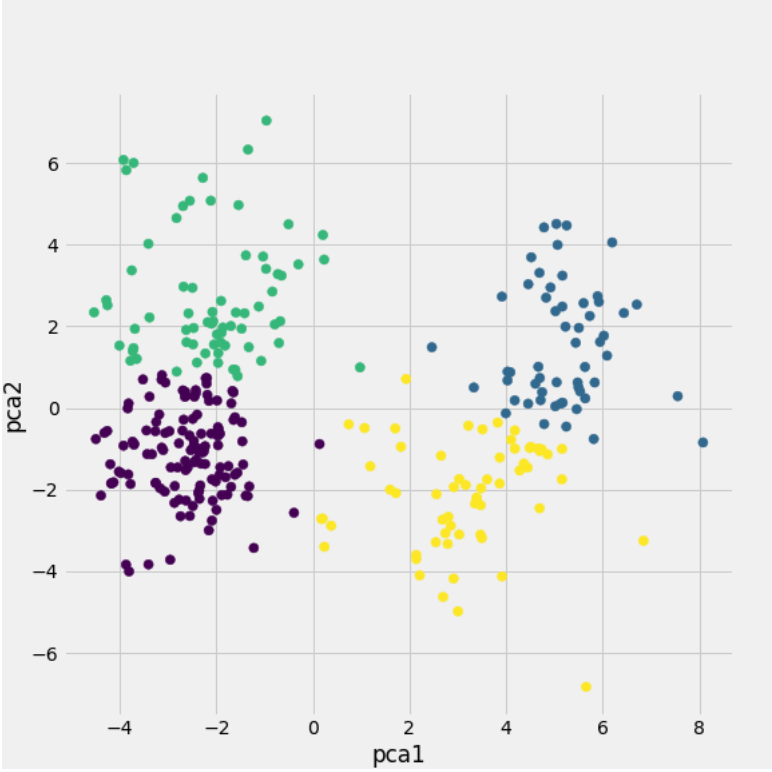


Figura 4.4: Clusters generados mediante el algoritmo de K-means.

Las características de los clusters son las siguientes.

En el cluster 0 nos podremos encontrar al conjunto de individuos más grande con 126. Estos individuos son los más delgados (medida de la cintura y cadera más pequeña) y además son los individuos que menos peso tienen con respecto a los otros clusters y a la media global. Disponen de valores de grasa corporal y musculatura por debajo de la media global y por debajo de los valores de los otros clusters.

Para el cluster 1, que está conformado por 55 individuos, tenemos que la media de sus pesos es de 109 Kg (está por encima de la media del total de pacientes). La medida de la cadera y cintura también están por encima de la media. Son individuos altos, entre 170 cm y 183 cm. Sus valores de colesterol son normales, pero los valores de glucosa están por debajo de lo normal y los triglicéridos son muy altos.

El cluster 2 está conformado por 70 individuos, los cuales son más bajos que los del cluster 1, entre 149 cm y 174 cm. Tienen valores de grasa corporal, en cada una de las partes del cuerpo por individual y de forma global, por encima que los individuos del cluster 1. en cambio, su musculatura es inferior en todos los sentidos a los del Cluster 1, aunque sus valores de musculatura están muy cerca de la media de todos los individuos.

En el cluster 3 nos encontramos con 61 individuos que disponen de valores de musculatura por debajo de la media global en la zona de los brazos, pero en las piernas están muy por encima de la media y de los otros clusters. Las medidas de la cadera y cintura de estos individuos está por debajo de la media.

5 Conclusiones

En este capítulo se resaltan las principales conclusiones obtenidas como consecuencia del trabajo de minería de datos desarrollado. También se detallan las líneas futuras derivadas del trabajo.

5.1. Conclusión

En el presente trabajo se han planteado una serie de objetivos relacionados con el análisis de datos morfológicos del cuerpo humano. El objetivo general ha sido abordado desde el desarrollo de los objetivos específicos.

En relación con el desarrollo de un sistema capaz de obtener datos médicos y geométricos se ha concluido que, se ha incluido funcionalidades en el software de Tech4Diet para el el almacenamiento de los datos médicos recopilados de diversas fuentes en una base de datos. Se han desarrollado en Python un conjunto de funciones encargadas de la creación de templates y obtención de datos geométricos 3D del cuerpo humano, aunque el estado actual de los templates generados mediante un registro deformable no permite la captación de medidas de forma automática.

Por lo que se refiere al trabajo de preprocesado de los datos se han aplicado satisfactoriamente diversas técnicas para la identificación de valores atípicos, para la imputación

de datos sobre valores nulos y se han filtrado todos los conjuntos de datos para que prevalezca la privacidad de los pacientes.

En relación al análisis de los datos, se han probado un total de diez algoritmos de regresión distintos y se ha comprobado su precisión con diversas métricas. Aquellos modelos con mejores resultados han sido optimizados con la finalidad de mejorar los resultados predichos. También se ha analizado que características son útiles para los agrupamientos de los pacientes utilizando técnicas de clustering.

5.2. Líneas futuras

El desarrollo de este trabajo a derivado en una serie de líneas futuras entre las que podemos destacar:

Mejora en la obtención de templates a partir de modelos 3D de cuerpos humanos como podría ser la utilización de técnicas de creación de templates actuales como SMPL-X o STAR.

La ampliación mediante datos psicológicos permitiría un análisis más amplio sobre el cual podrían obtener conclusiones más precisas sobre como sacar deducciones más certeras de como afectan los tratamientos nutricionales, que utilizan tecnologías 3D para mejorar su adherencia, sobre las habilidades cognitivas del paciente.

La ampliación de sesiones o de pacientes permitiría ser capaces de obtener modelos más precisos de regresión.

6 Anexos

6.1. Gráficos de caja y de dispersión

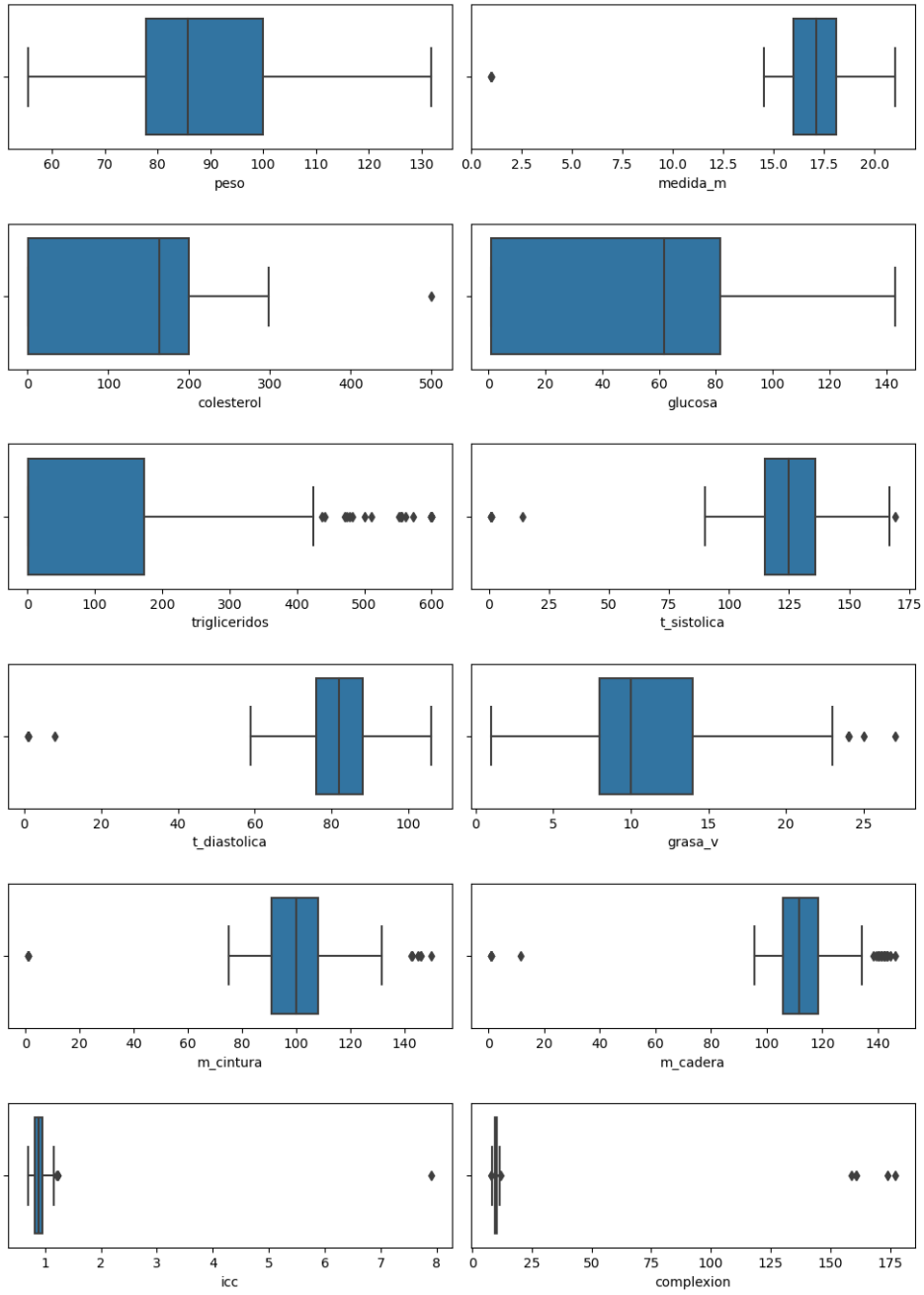


Figura 6.1: Gráfico de caja del dataframe sesión (parte 1).

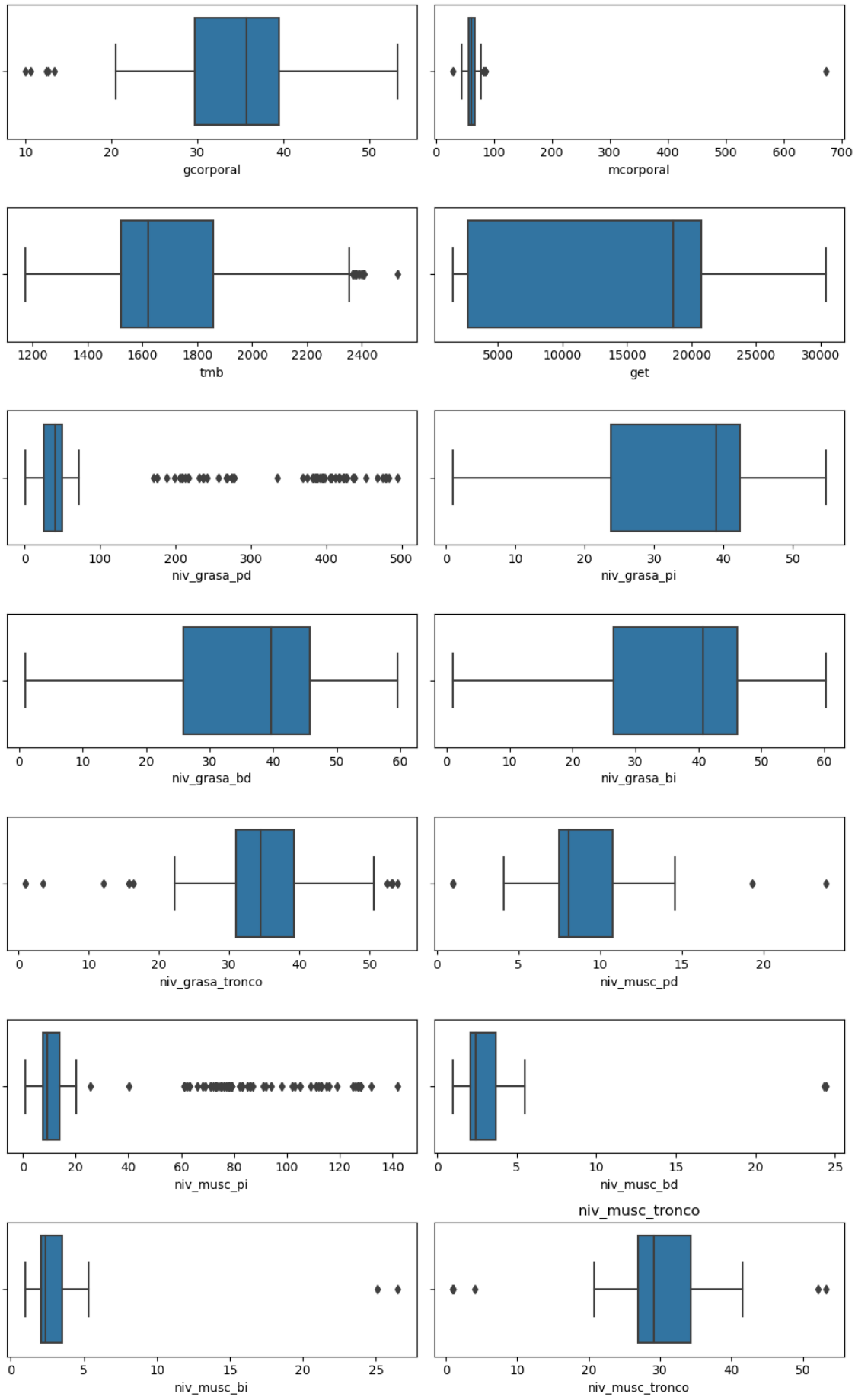


Figura 6.2: Gráfico de caja del dataframe sesión (parte 2).

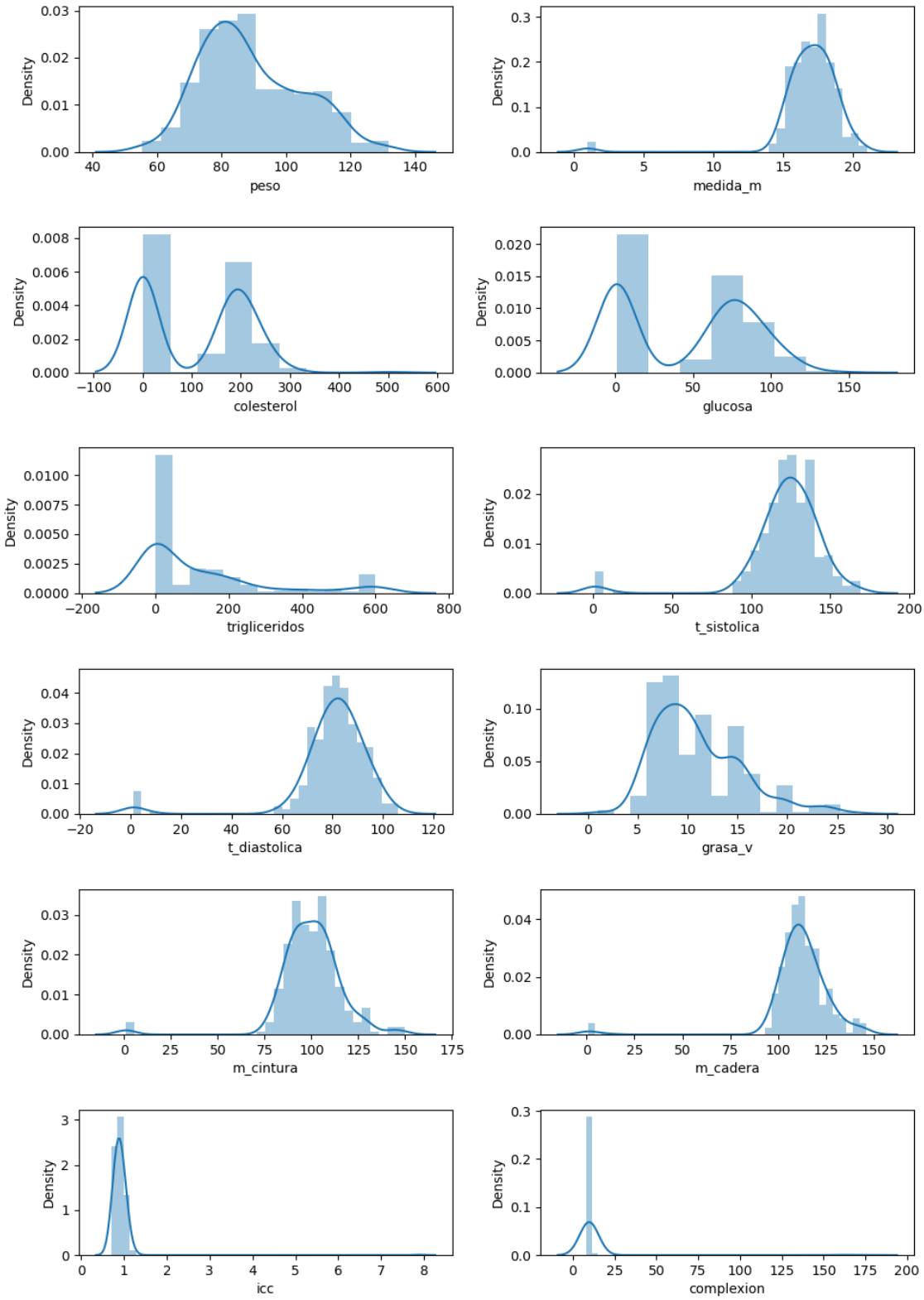


Figura 6.3: Gráfico de dispersión del dataframe sesión (parte 1).

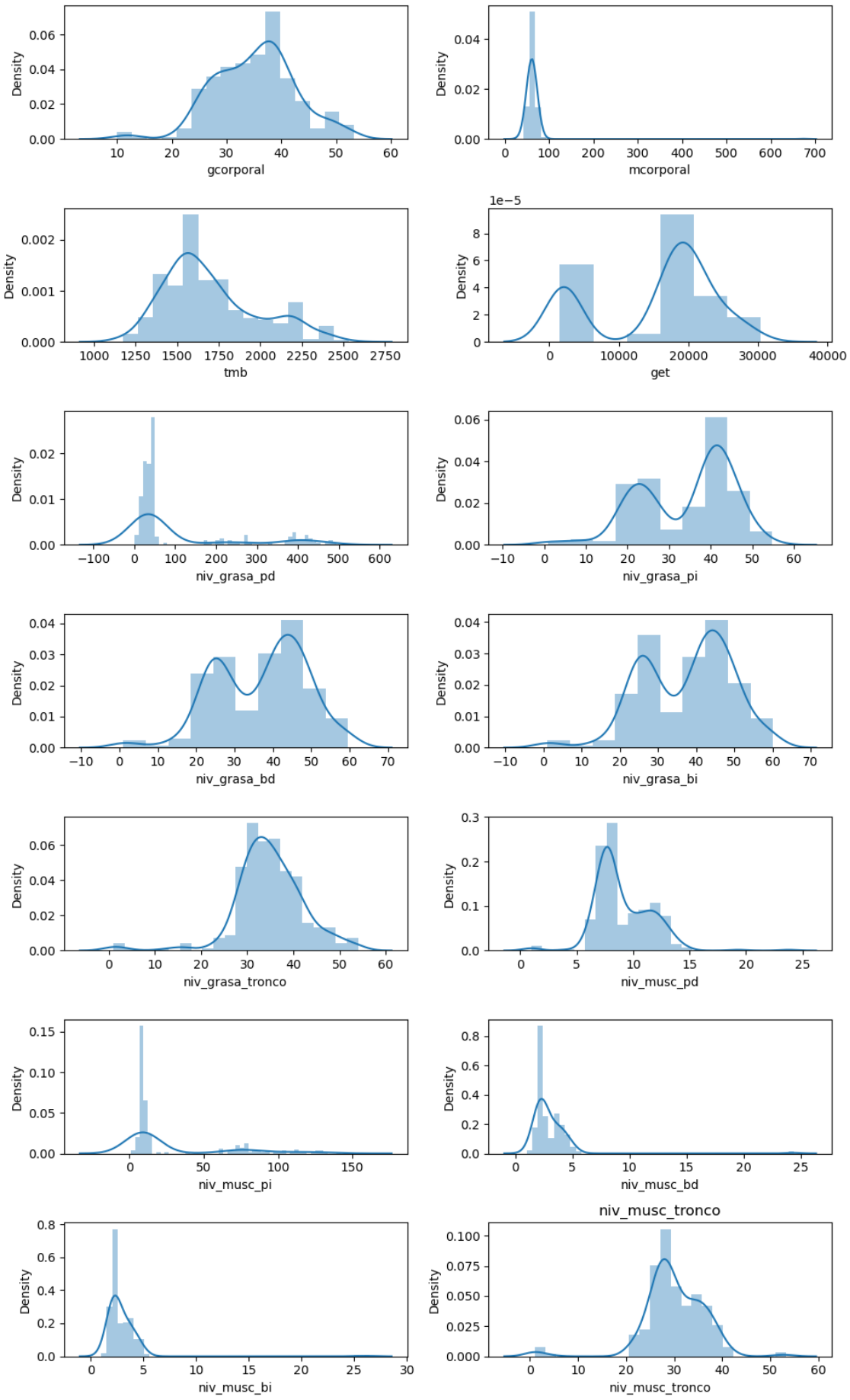


Figura 6.4: Gráfico de dispersión del dataframe sesión (parte 2).

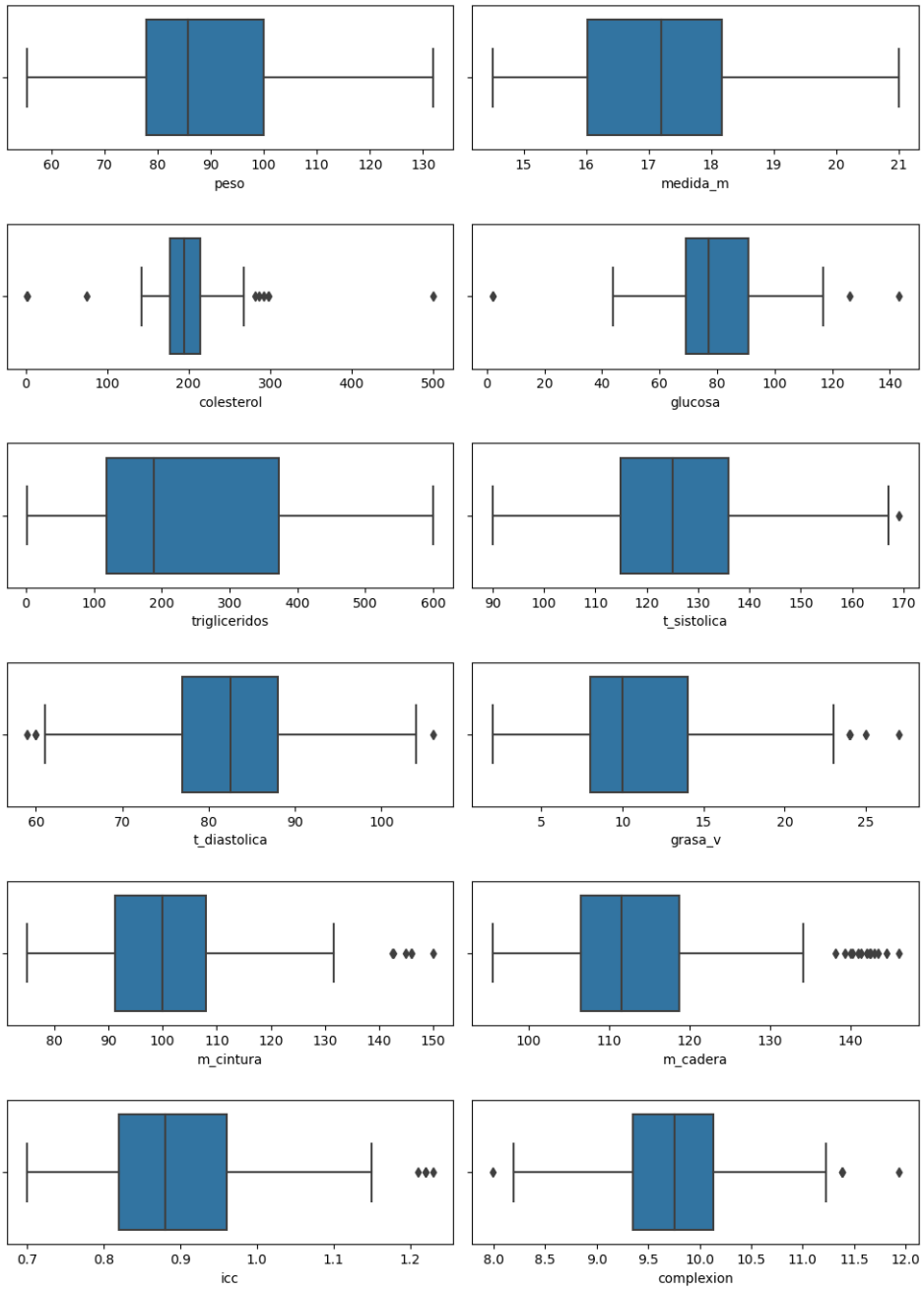


Figura 6.5: Gráfico de caja del dataframe sesión después de aplicar tareas de preprocesamiento (parte 1).

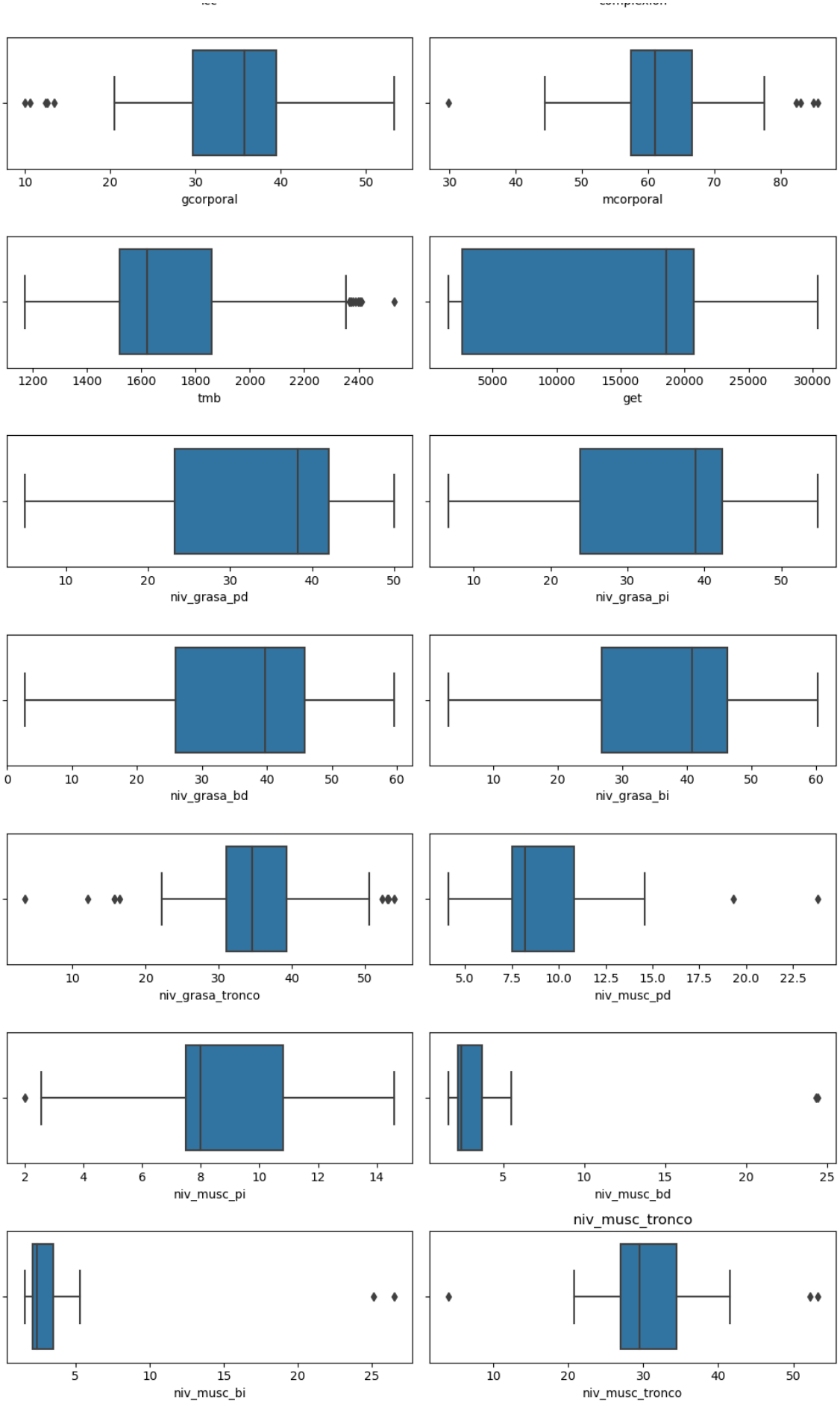


Figura 6.6: Gráfico de caja del dataframe sesión después de aplicar tareas de preprocesamiento (parte 2).

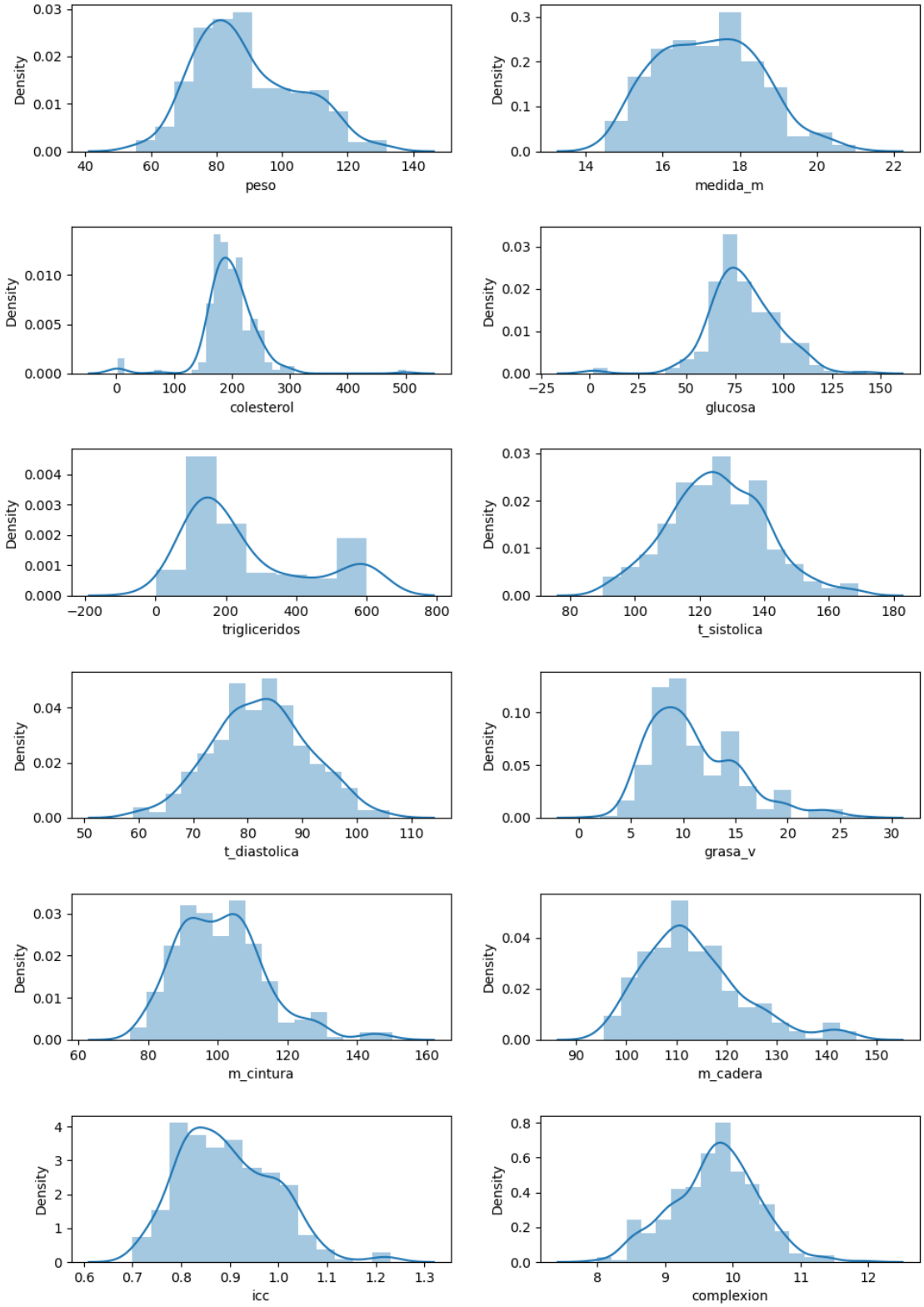


Figura 6.7: Gráfico de dispersión del dataframe sesión después de aplicar tareas de preprocesamiento (parte 1).

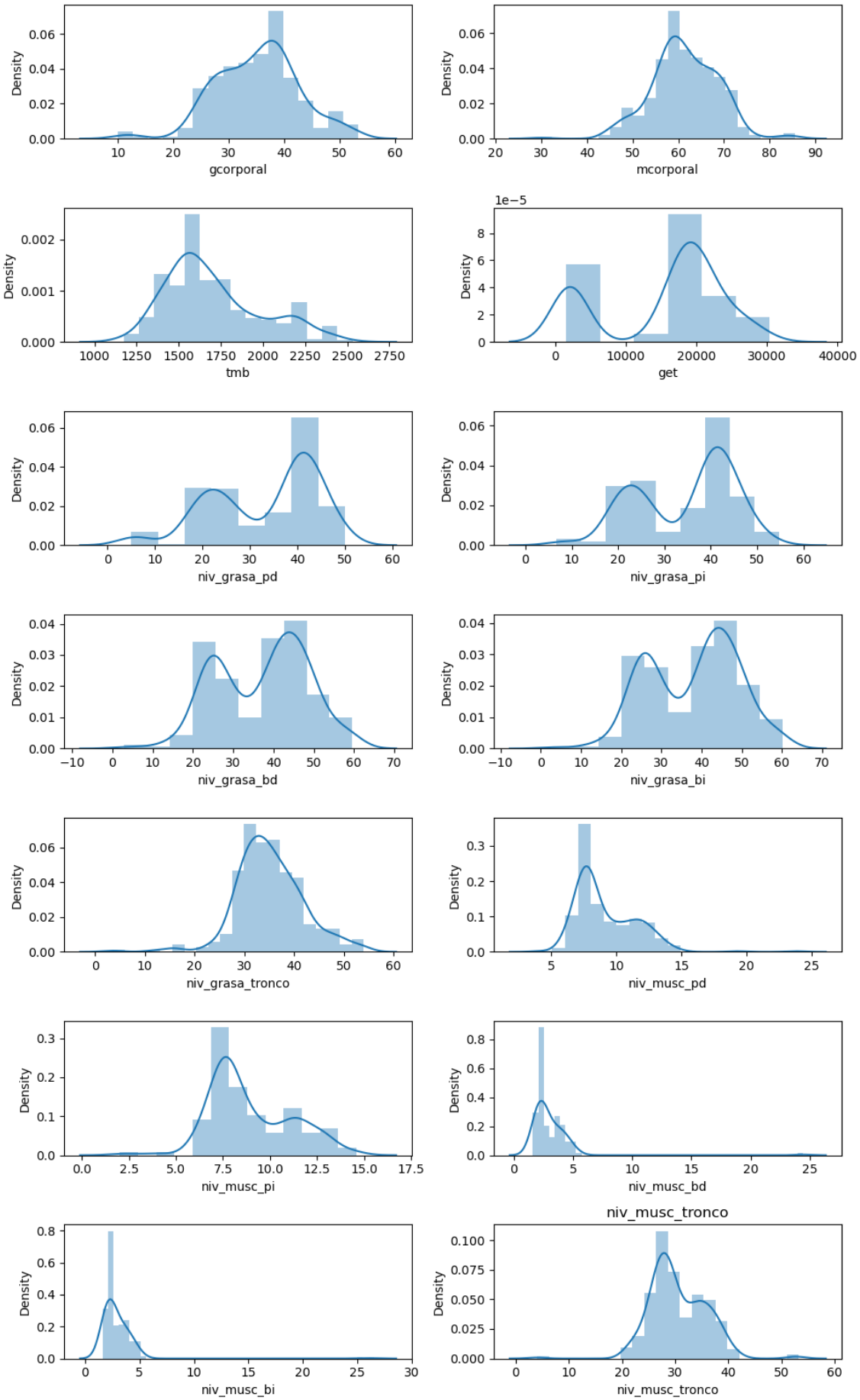


Figura 6.8: Gráfico de dispersión del dataframe sesión después de aplicar tareas de preprocesamiento (parte 2).

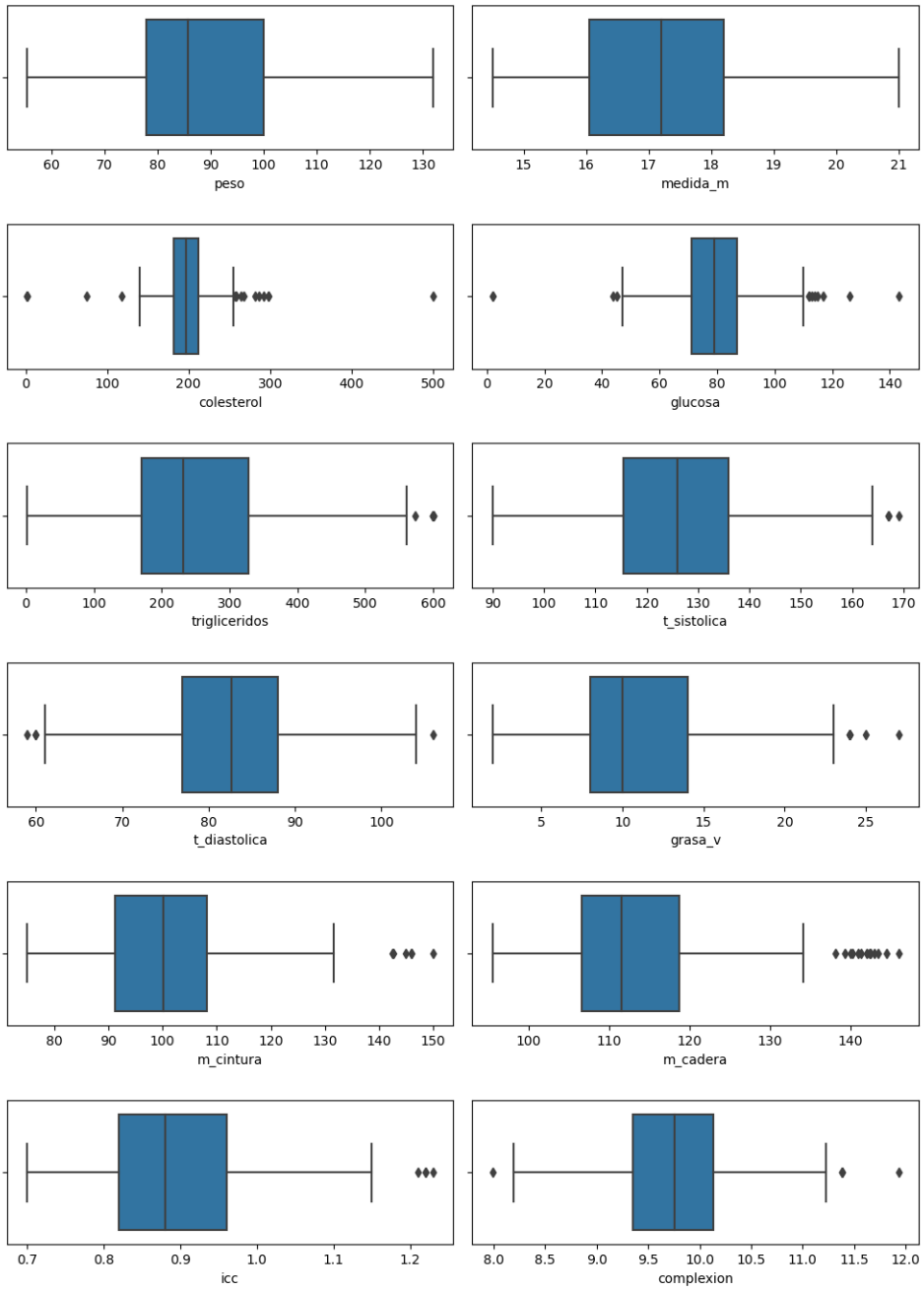


Figura 6.9: Gráfico de caja del dataframe sesión después de aplicar la imputación de datos (parte 1).

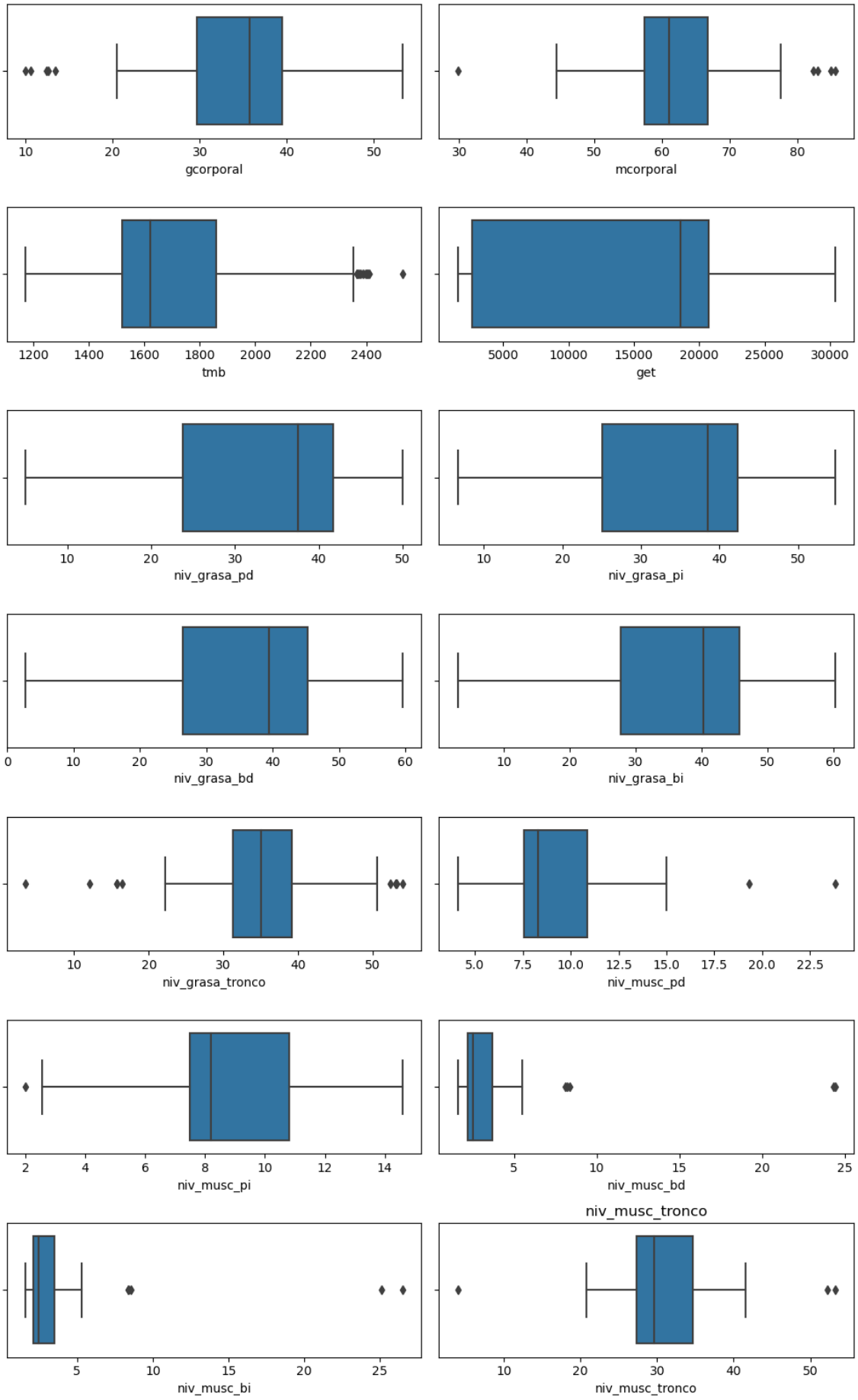


Figura 6.10: Gráfico de caja del dataframe sesión después de aplicar la imputación de datos (parte 2).

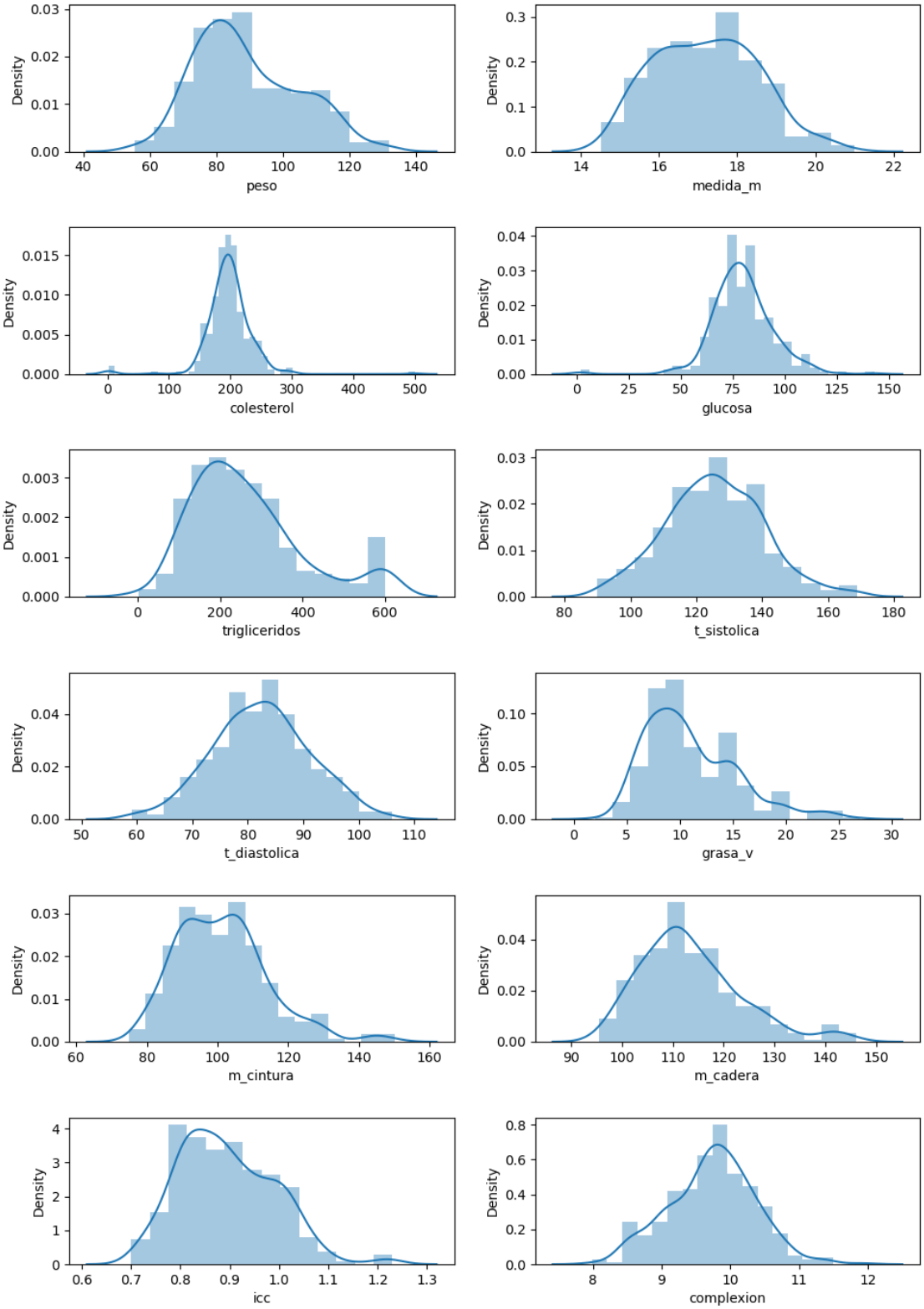


Figura 6.11: Gráfico de dispersión del dataframe sesión después de aplicar la imputación de datos (parte 1).

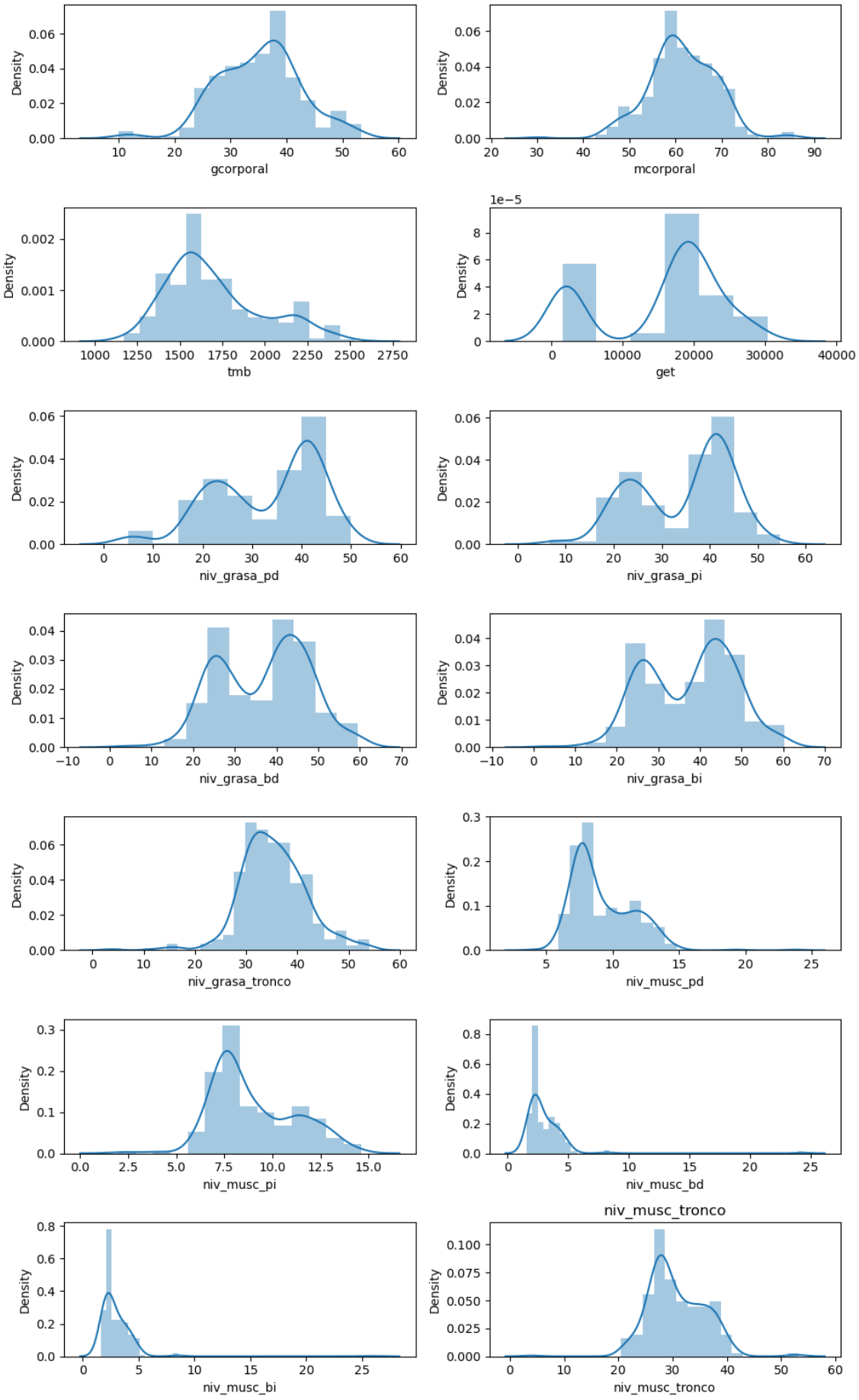
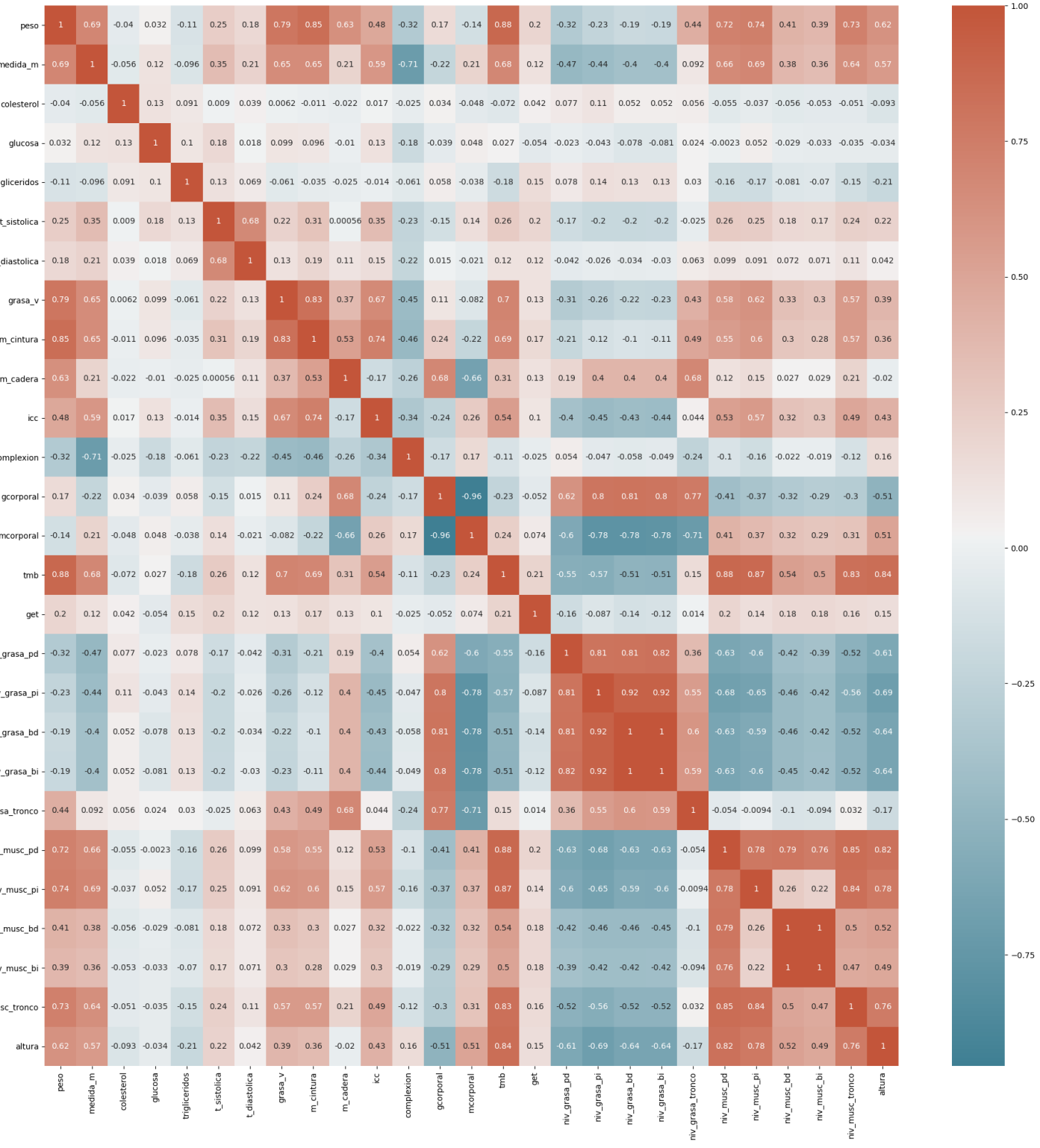


Figura 6.12: Gráfico de dispersión del dataframe sesión después de aplicar la imputación de datos (parte 2).

6.2. Correlaciones entre las variables de la tabla sesión



6.3. Tablas con los resultados de las métricas para los diferentes modelos regresores

Tabla 6.1: Métricas obtenidas para el conjunto de test y entrenamiento sobre las variables relacionadas con las medidas antropométricas para predecir colesterol

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	8.141547	115.340175	10.739654	0.772996
KNN	10.033945	174.355780	13.204385	0.656846
RFR	12.591742	281.654799	16.782574	0.445668
ETR	13.049502	309.207152	17.584287	0.391442
DTC	12.752118	327.707337	18.102689	0.355031
ABR	14.786477	333.559137	18.263601	0.343514
EN	16.259511	480.312927	21.916043	0.054684
Huber	16.297174	481.751777	21.948845	0.051852
LSVR	29.266554	1152.834684	33.953419	-1.268923
PAR	39.555430	2108.798090	45.921652	-3.150378

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR	16.519192	533.272239	23.092688	0.123721
GBR	16.972546	546.769806	23.383109	0.101541
RFR	16.762584	549.067470	23.432189	0.097766
ABR	18.150473	588.256280	24.253995	0.033370
KNN	18.264894	625.676489	25.013526	-0.028119
EN	18.509753	633.258265	25.164623	-0.040578
Huber	18.585772	646.849208	25.433230	-0.062910
DTC	18.052051	650.495960	25.504822	-0.068903
LSVR	28.201995	1120.432727	33.472866	-0.841108
PAR	41.802853	2399.683863	48.986568	-2.943189

Tabla 6.2: Métricas obtenidas para el conjunto de test y entrenamiento con el conjunto de datos de musculatura para predecir el colesterol

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	7.575941	93.670813	9.678368	0.815644
KNN	9.761468	158.823578	12.602523	0.687415
RFR	11.051174	205.591305	14.338455	0.595371
DTC	11.614627	230.295652	15.175495	0.546749
ABR	13.168343	256.948517	16.029614	0.494293
ETR	12.238912	281.392160	16.774748	0.446185
EN	16.246980	471.169874	21.706448	0.072679
Huber	16.362155	473.315949	21.755826	0.068455
LSVR	18.875397	635.159761	25.202376	-0.250074
PAR	38.628256	2021.236748	44.958167	-2.978046

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR	16.963503	530.863675	23.040479	0.127678
KNN	17.689362	548.167660	23.412981	0.099244
RFR	17.521455	567.531942	23.822929	0.067425
GBR	17.864909	579.337088	24.069422	0.048026
EN	18.244322	606.914397	24.635633	0.002711
DTC	19.144935	616.403677	24.827478	-0.012882
ABR	18.614396	622.378556	24.947516	-0.022700
Huber	18.454267	626.489398	25.029770	-0.029455
LSVR	19.456770	743.676095	27.270425	-0.222017
PAR	41.092945	2315.908709	48.123889	-2.805529

Tabla 6.3: Métricas obtenidas para el conjunto de test y entrenamiento con el conjunto de datos de grasa corporal para predecir el colesterol

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	7.417494	93.325983	9.660537	0.816323
KNN	10.816514	191.815505	13.849747	0.622483
RFR	11.880173	233.194123	15.270695	0.541045
DTC	11.923220	258.656081	16.082788	0.490933
ABR	13.805454	286.750084	16.933697	0.435640
ETR	13.469433	327.965560	18.109819	0.354523
EN	16.599646	483.248421	21.982912	0.048907
Huber	16.563402	484.248694	22.005651	0.046938
LSVR	25.979077	1085.275624	32.943522	-1.135958
PAR	40.548782	2457.906092	49.577274	-3.837466

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR	18.271084	603.948783	24.575369	0.007584
EN	18.308279	608.665704	24.671151	-0.000167
Huber	18.335775	615.296038	24.805162	-0.011062
RFR	18.904361	643.079543	25.359013	-0.056716
GBR	19.302524	657.763530	25.646901	-0.080845
ABR	19.141193	666.427831	25.815264	-0.095082
KNN	20.477660	745.261383	27.299476	-0.224622
DTC	20.856000	772.274481	27.789827	-0.269011
LSVR	29.059434	1298.662573	36.036961	-1.133978
PAR	40.694991	2418.909476	49.182410	-2.974781

Tabla 6.4: Métricas obtenidas para el conjunto de test y entrenamiento con todas las variables para predecir el colesterol

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	5.290304	46.486460	6.818098	0.908509
RFR	10.023204	168.190656	12.968834	0.668980
KNN	8.762385	171.051560	13.078668	0.663349
DTC	10.052128	186.921512	13.671924	0.632115
ABR	12.198567	214.885696	14.658980	0.577078
ETR	11.058388	230.018840	15.166372	0.547294
EN	14.806472	386.140300	19.650453	0.240028
Huber	15.797157	485.245447	22.028287	0.044976
LSVR	16.828659	521.117669	22.828002	-0.025625
PAR	43.891621	2570.590645	50.700993	-4.059243

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR	15.804693	496.637213	22.285359	0.183920
KNN	14.838298	527.906809	22.976223	0.132537
ETR	16.667783	535.255784	23.135596	0.120461
EN	17.494553	548.797207	23.426421	0.098210
DTC	18.022558	566.926520	23.810219	0.068420
GBR	17.193845	571.437517	23.904759	0.061007
ABR	18.036508	602.470313	24.545271	0.010014
Huber	18.320528	728.350211	26.987964	-0.196834
LSVR	19.216837	805.037412	28.373181	-0.322847
PAR	46.552175	2909.287410	53.937811	-3.780576

Tabla 6.5: Métricas obtenidas para el conjunto de test y entrenamiento sobre las variables relacionadas con las medidas antropométricas para predecir predecir glucosa

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	4.975282	40.840650	6.390669	0.746293
KNN	5.999083	59.576789	7.718600	0.629901
RFR	6.930926	78.959878	8.885937	0.509491
ABR	7.846181	90.694997	9.523392	0.436591
ETR	7.350587	92.104640	9.597116	0.427834
DTC	7.077422	92.146466	9.599295	0.427574
EN	9.603975	151.308629	12.300757	0.060051
Huber	9.571404	151.854820	12.322939	0.056658
PAR	13.865229	269.053111	16.402839	-0.671393
LSVR	23.987502	729.971266	27.017980	-3.534676

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR	9.259795	155.911968	12.486471	0.052514
ABR	9.641203	159.150011	12.615467	0.032837
EN	9.473186	161.852592	12.722130	0.016413
Huber	9.639267	167.119827	12.927483	-0.015596
RFR	9.730302	169.858694	13.032985	-0.032241
KNN	10.627660	186.755106	13.665837	-0.134921
GBR	10.278759	195.631169	13.986821	-0.188861
DTC	11.145711	231.046895	15.200227	-0.404085
PAR	13.004743	249.983101	15.810854	-0.519161
LSVR	24.347788	716.984789	26.776572	-3.357156

Tabla 6.6: Métricas obtenidas para el conjunto de test y entrenamiento con el conjunto de datos de musculatura para predecir la glucosa

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	4.628535	37.323642	6.109308	0.768141
KNN	6.439450	71.696055	8.467352	0.554615
RFR	7.103792	84.880353	9.213053	0.472712
ABR	8.217159	99.044265	9.952098	0.384724
DTC	7.422640	101.018079	10.050775	0.372463
ETR	7.849367	107.742297	10.379899	0.330691
EN	9.574809	149.888801	12.242908	0.068871
Huber	9.551356	154.520825	12.430641	0.040097
LSVR	9.808744	160.457179	12.667169	0.003219
PAR	20.812172	561.603874	23.698183	-2.488756

	MAE Test	MSE Test	RMSE Test	R_Squared Test
ETR	9.149787	154.528772	12.430960	0.060920
RFR	9.076373	156.072258	12.492888	0.051540
ABR	9.034499	156.797337	12.521874	0.047134
DTC	9.188469	159.401701	12.625439	0.031307
EN	9.377430	165.430250	12.861969	-0.005329
Huber	9.337640	167.125527	12.927704	-0.015631
GBR	9.641658	170.218047	13.046764	-0.034424
KNN	10.386170	180.423298	13.432174	-0.096442
LSVR	9.783845	180.697378	13.442372	-0.098108
PAR	20.982226	572.814992	23.933554	-2.481028

Tabla 6.7: Métricas obtenidas para el conjunto de test y entrenamiento con el conjunto de datos de grasa corporal para predecir la glucosa

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	4.151027	29.157226	5.399743	0.818872
KNN	6.660092	73.056927	8.547334	0.546161
RFR	7.042230	80.464489	8.970200	0.500144
DTC	7.415679	104.031515	10.199584	0.353743
ETR	8.167804	108.698428	10.425854	0.324751
ABR	8.807401	109.610380	10.469498	0.319086
EN	9.756151	156.691964	12.517666	0.026609
Huber	9.693871	157.283320	12.541265	0.022936
LSVR	11.998337	243.014059	15.588908	-0.509635
PAR	25.375724	835.288301	28.901355	-4.188919

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR	9.356822	160.419306	12.665674	0.025123
EN	9.370868	162.749386	12.757327	0.010963
ETR	9.610032	165.465401	12.863336	-0.005542
Huber	9.452848	166.446104	12.901399	-0.011502
DTC	9.409764	168.154537	12.967441	-0.021884
ABR	9.835421	169.272191	13.010465	-0.028676
GBR	10.200251	186.496398	13.656368	-0.133349
KNN	11.289362	205.409787	14.332124	-0.248287
LSVR	12.889823	260.665340	16.145134	-0.584078
PAR	24.007384	733.729140	27.087435	-3.458912

Tabla 6.8: Métricas obtenidas para el conjunto de test y entrenamiento con todas las variables para predecir la glucosa

	MAE Train	MSE Train	RMSE Train	R_Squared Train
GBR	2.712431	11.734698	3.425595	0.927103
KNN	4.787615	51.641239	7.186184	0.679198
RFR	6.022263	62.200921	7.886756	0.613600
ABR	6.518580	62.471831	7.903912	0.611917
ETR	6.447717	73.683163	8.583890	0.542271
DTC	6.361737	77.882891	8.825128	0.516181
EN	8.878780	127.941110	11.311106	0.205213
Huber	9.213834	153.056163	12.371587	0.049195
LSVR	13.980155	318.085393	17.834949	-0.975987
PAR	21.985059	661.748566	25.724474	-3.110867

	MAE Test	MSE Test	RMSE Test	R_Squared Test
RFR	8.273479	129.229875	11.367932	0.214663
GBR	8.605590	137.797511	11.738718	0.162597
ABR	8.784760	141.361764	11.889565	0.140937
ETR	8.532709	141.500524	11.895399	0.140094
EN	9.273577	155.696161	12.477827	0.053826
DTC	9.567408	167.672584	12.948845	-0.018955
KNN	9.482979	183.162340	13.533748	-0.113088
Huber	10.372840	185.383232	13.615551	-0.126584
LSVR	13.935975	291.355922	17.069151	-0.770586
PAR	20.807917	586.281028	24.213241	-2.562862

Bibliografía

[azu, 2021] (2021). *Azure Machine Learning*.
`https://azure.microsoft.com/es-es/services/machine-learning/` (accessed Ago. 2021).

[ibm, 2021] (2021). *IBM SPSS Modeler*.
`https://www.ibm.com/es-es/products/spss-modeler` (accessed Ago. 2021).

[kee, 2021] (2021). *Keel*.
`http://www.keel.es/` (accessed Ago. 2021).

[kni, 2021] (2021). *Knime*.
`https://www.knime.com/` (accessed Ago. 2021).

[Ora, 2021] (2021). *Orange*.
`https://orangedatamining.com/` (accessed Ago. 2021).

[r, 2021] (2021). *The R Project for Statistical Computing*.
`https://www.r-project.org/` (accessed Ago. 2021).

[Rap, 2021] (2021). *Rapidminer*.
`https://rapidminer.com/` (accessed Ago. 2021).

[Wek, 2021] (2021). *Weka*.

`https://www.cs.waikato.ac.nz/ml/weka/` (accessed Ago. 2021).

[Aleixandre-Tudó et al., 2016] Aleixandre-Tudó, J., Álvarez, I., García, M. J., Lizama, V., and Aleixandre, J. L. (2016). Application of multivariate regression methods to predict sensory quality of red wines. *Czech Journal of Food Sciences*, 33:217–227.

[Andrienko and Andrienko, 2005] Andrienko, N. and Andrienko, G. (2005). Exploratory analysis of spatial and temporal data. a systematic approach.

[Bogo et al., 2016] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016). Keep it smpl: Automatic estimation of 3d human pose and shape from a single image.

[Ester et al.,] Ester, M., Kriegel, H. P., Sander, J., and Xiaowei, X. A density-based algorithm for discovering clusters in large spatial databases with noise.

[Fayyad et al., 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 82–88. AAAI Press.

[Fuster-Guilló et al., 2021] Fuster-Guilló, A., Azorín-López, J., Castillo-Zaragoza, J. M., Manchón-Pernis, C., Pérez-Pérez, L. F., and Zaragoza-Martí, A. (2021). Multidimensional measurement of virtual human bodies acquired with depth sensors. In Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., and Corchado, E., editors, *15th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2020)*, pages 721–730, Cham. Springer International Publishing.

[Fuster-Guilló et al., 2020] Fuster-Guilló, A., Jorge Azorín-López, M. S.-C., Castillo-Zaragoza, J. M., Garcia-D'Urso, N., and Fisher, R. B. (2020). *RGB-D based framework to Acquire, Visualize and Measure the Human Body for Dietetic Treatments*.

[Garland and Heckbert, 1997] Garland, M. and Heckbert, P. (1997). Surface simplification using quadric error metrics. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 1997.

[Ge et al., 2017] Ge, Z., Song, Z., Ding, S. X., and Huang, B. (2017). Data mining and analytics in the process industry: The role of machine learning. *IEEE Access*, 5:20590–20616.

[Hosseini, 2021] Hosseini, S., S.-S. (2021). *Data mining tools -a case study for network intrusion detection.*

[Jovic et al., 2014] Jovic, A., Brkić, K., and Bogunovic, N. (2014). An overview of free software tools for general data mining. pages 1112–1117.

[KDnuggets, 2021] KDnuggets (2021). *Meetings 2003 KDnuggets.* `https://www.kdnuggets.com/meetings-past/past-meetings-2003.html` (accessed Ago. 2021).

[Koh and Tan, 2005] Koh, H. and Tan, G. (2005). Data mining applications in healthcare. *Journal of healthcare information management : JHIM*, 19:64–72.

[Kovalerchuk and Vityaev, 2005] Kovalerchuk, B. and Vityaev, E. (2005). *Data Mining for Financial Applications*, pages 1203–1224.

[Kvålseth, 1985] Kvålseth ((1985).). Cautionary Note About R2. page 279–285.

[Li et al., 2017] Li, T., Bolkart, T., Black, M., Li, H., and Romero, J. (2017). Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36:1–17.

[Liao et al., 2012] Liao, S.-H., Chu, P.-H., and Hsiao, P.-Y. (2012). Data mining techniques and applications – a decade review from 2000 to 2011. *Expert Systems with*

Applications, 39(12):11303–11311.

[Loper et al., 2015] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34:248:1–248:16.

[Miao et al., 2017] Miao, F., Fu, N., Zhang, Y.-T., Ding, X.-R., Hong, X., He, Q., and Li, Y. (2017). A novel continuous blood pressure estimation approach based on data mining techniques. *IEEE Journal of Biomedical and Health Informatics*, 21(6):1730–1740.

[Mokharrak et al., 2012] Mokharrak, W., Al Khalaf, N., and Altman, T. (2012). *Application of Bioinformatics and Data Mining in Cancer Prediction.*

[Open3D, 2021] Open3D (2021). *Open3d Geometry Simplify Quadric Decimation.* `http://www.open3d.org/docs/` (accessed Ago. 2021).

[Pandas, 2021] Pandas (2021). *Pandas (software).* `https://pandas.pydata.org/` (accessed Ago. 2021).

[Pavlakos et al., 2019] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. (2019). Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985.

[Provost and Fawcett, 2013] Provost, F. and Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking.* O'Reilly Media, Inc., 1st edition.

[Romero et al., 2017] Romero, J., Tzionas, D., and Black, M. (2017). Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36.

[Rosas and Verdejo, 2009] Rosas, J. and Verdejo, E. (2009). Métodos de imputación para el tratamiento de datos faltantes: aplicación mediante r/splus = imputation methods to handle the problem of missing data: an application using r/splus. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, 7.

[Rubin and Service, 1978] Rubin, D. B. and Service, E. T. (1978). Multiple imputations in sample surveys - a phenomenological bayesian approach to nonresponse.

[Saeb et al., 2018] Saeb, A., David, S., Rafiullah, M., and Al-Rubeaan, K. (2018). *Classification Techniques and Data Mining Tools Used in Medical Bioinformatics*, pages 105–126.

[Scikit-learn, 2021] Scikit-learn (2021). *KNNImputer*. `https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html` (accessed Ago. 2021).

[Tech4Diet, 2019a] Tech4Diet (2019a). *Project TIN2017-89069-R Spanish State Research Agency (AEI). 4D modelling and visualization of the human body to improve adherence to dietetic-nutritional intervention of obesity.* `http://tech4d.dtic.ua.es/` (accessed Nov. 2019).

[Tech4Diet, 2019b] Tech4Diet, M. 2019b). *Project TIN2017-89069-R Spanish State Research Agency (AEI). 4D modelling and visualization of the human body to improve adherence to dietetic-nutritional intervention of obesity.* `http://tech4d.dtic.ua.es/media/` (accessed Ago. 2021).

[Tomasevic and Vraneš, 2019] Tomasevic, Nikola, G. N. and Vraneš, S. (2019). *An overview and comparison of supervised data mining techniques for student exam performance prediction.*

[Trimesh, 2021] Trimesh (2021). *Trimeh Registration.* `https://trimsh.org/trimesh.registration` (accessed Ago. 2021).

[Witten et al., 2011] Witten, I. H., Frank, E., and Hall, M. A. ((2011).). Data Mining-Practical Machine Learning Tools and Techniques.

[XH et al., 2014] XH, Z., C, Z., D, L., and X., D. (2014). Applied missing data analysis in the health sciences.

[Zelenkov and Anissichkina, 2021] Zelenkov, Y. and Anissichkina, E. (2021). Trends in data mining research: A two-decade review using topic analysis. *Business Informatics*, 15:30–46.

[Zhang and Zhang, 2009] Zhang, Y. and Zhang, Y. (2009). Complex process monitoring using modified partial least squares method of independent component regression. *Chemometrics and Intelligent Laboratory Systems*, 98:143–148.

[Álvaro Jiménez Galindo, 2010] Álvaro Jiménez Galindo, H. G. (2010). *Minería de Datos en la Educación.*